Oracle Fusion Cloud Sales Automation

How do I configure the Sales Dashboard in the Redwood User Experience?

Oracle Fusion Cloud Sales Automation
How do I configure the Sales Dashboard in the Redwood User Experience?

F90641-13

Author: Jiri Weiss

Contents

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

Some application pages have help icons ⓘ to give you access to contextual help. 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. If the page has contextual help, help icons will appear.

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 Introduction and Setup Overview

How to Use this Playbook

This playbook provides the steps sales administrators need to configure sales dashboards for their organization. It also includes information to help salespeople personalize their dashboards using the content the administrators provide. Administrators can configure any number of sales dashboards for different audiences with the content provided by Oracle and with the content they create.

1. Familiarize yourself with the types of information you can display and the prerequisites.
2. Enable navigation to the Sales Dashboard from the Home page as described in the topic: *Enable Navigation to the Sales Dashboard*.
3. To configure your dashboards, follow the order of steps in the setup overview provided in the topic: *Overview of Sales Dashboard Configuration Steps*.
4. Salespeople can read the *Personalize Your Sales Dashboard* topic to learn how to personalize their dashboards.

The FAQ chapters provide common troubleshooting information.

Note:

- After you create your own custom versions of sales dashboard features, the underlying code provided by Oracle in those features is no longer automatically updated. Any updates are your organization's responsibility.
- Oracle Visual Builder Studio (VB Studio) is a full-blown development environment. This playbook doesn't cover all of the steps required to set up or use VB Studio. For more information on using VB Studio, see the *Extending Oracle Cloud Applications with Visual Builder Studio* guide.

Overview of the Sales Dashboard in the Redwood UX

The Sales Dashboard displays key sales performance metrics and key performance indicators (KPIs) that provide the sales organization with quick and easy-to-understand overviews of a sales team's to-do list and pipeline. Sales team members and managers can review items that need immediate attention, such as upcoming appointments and overdue tasks.

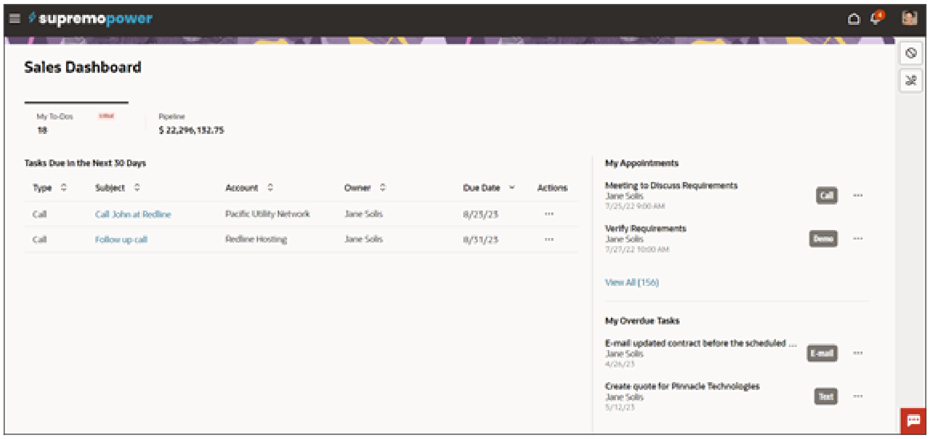
From the dashboard, a salesperson can drill-down into individual items, view all items by clicking View All, and take actions using the Actions menu.

Oracle provides you with two preconfigured dashboards that you can copy and configure.

Sales Dashboard for the Sales Representative

The dashboard includes two tabs:

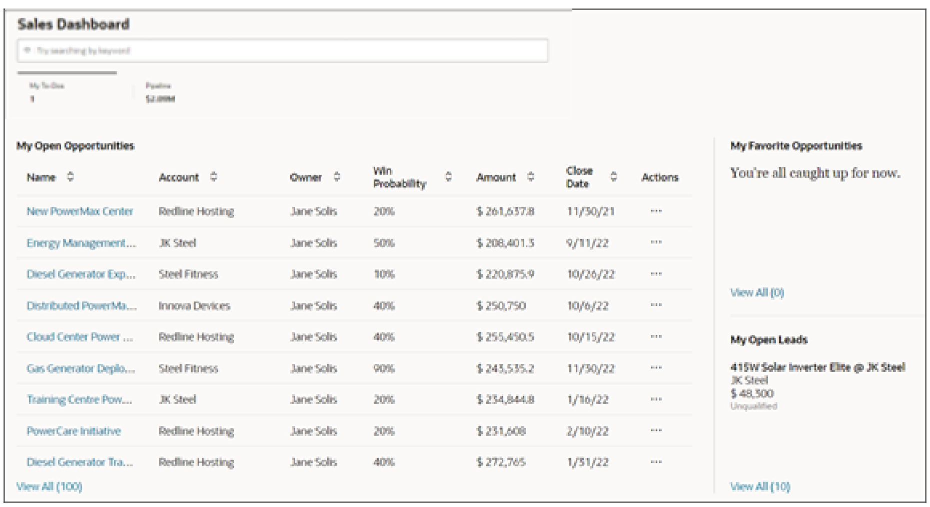
- **My To-Dos**



The My To-Dos tab includes these components:

- Tasks Due in the Next 30 Days table
- My Appointments list
- My Overdue Tasks list

- **Pipeline**



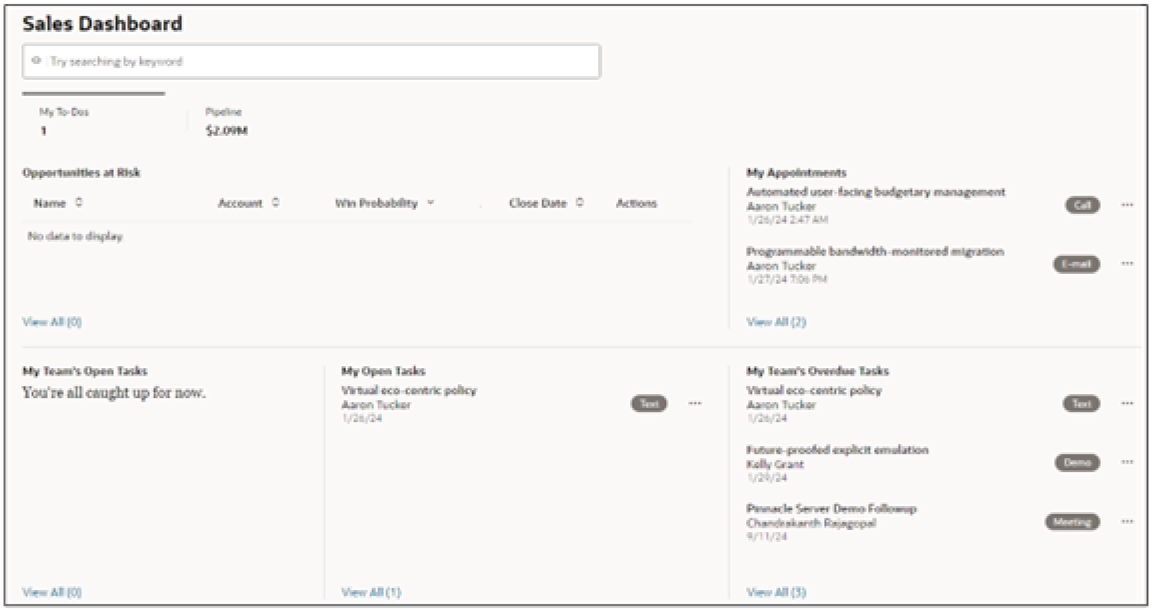
The Pipeline tab includes these components:

- My Open Opportunities table
- My Favorite Opportunities list
- My Open Leads list

Sales Dashboard for the Sales Manager

The sales manager dashboard includes the same two tabs, but tailored to managers:

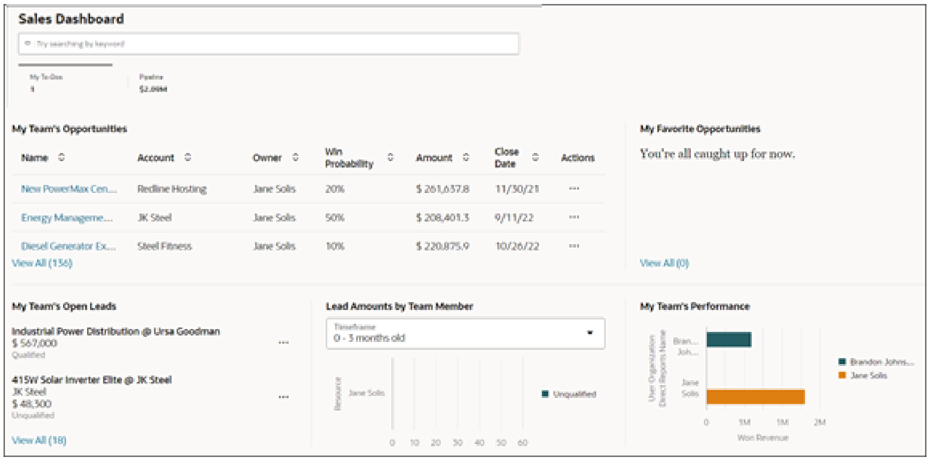
- **To-Dos**



The sales manager To-Dos tab includes these components all drawn from saved searches:

- Opportunities at Risk table
- My Appointments list
- My Team's Open Tasks list
- My Open Tasks list
- My Team's Overdue Tasks list

- **Pipeline**



The Pipeline page for the sales manager includes these components:
   - My Team's Opportunities table
   - My Favorite Opportunities list
   - My Team's Open Leads list
   - Lead Amounts by Team Member visualization
   - My Team's Performance visualization

Sales Dashboard VB Studio Component Basics

Here's a quick overview of how the Sales Dashboard is represented in Oracle Visual Builder Studio components.

Metrics Container Instances: The Shell of Your Dashboard

In Oracle Visual Builder Studio, the collection of tabs that make up a Sales Dashboard is represented by Metrics Container instances (cases).

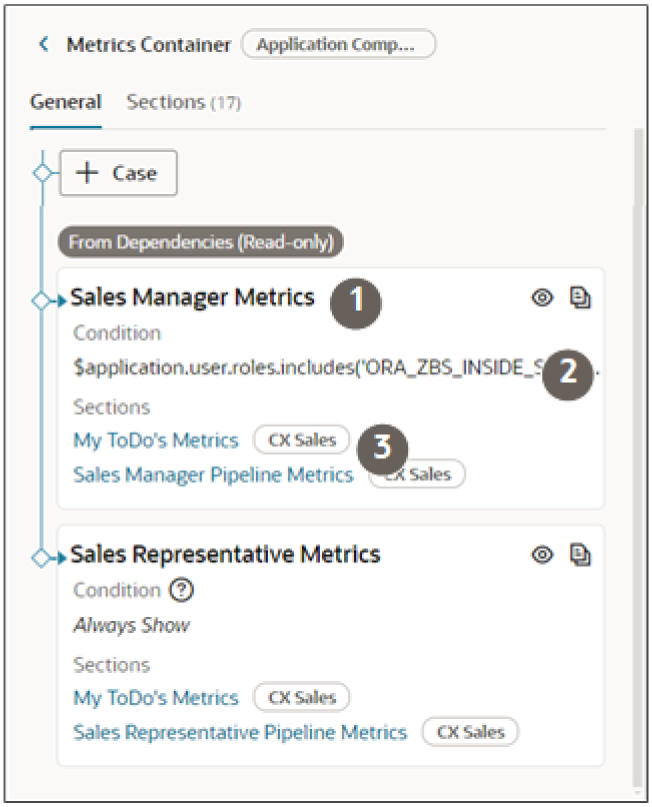
Each metrics container instance is composed of a list of Metric Cards under the Sections heading. Each metric card represents a tab on the dashboard. The order of the sections determines the order of the tabs.

The metrics container also includes a condition that determines who sees the dashboard.

Here's a screenshot of the two metric containers for the two sales dashboards provided by Oracle.

Callout Number	Description
1	The metrics container for the predefined sales manager dashboard.
2	Condition that specifies who sees this version of the dashboard. The predefined sales manager dashboard can be viewed only by the job roles listed. All other users see the sales representative dashboard because its condition is set to "Always Show".

Callout Number	Description
3	Metrics cards (the tabs in the dashboard) are listed under the Section heading. The order of the metric cards determines the order of the tabs. Metrics cards, layouts, and other elements predefined by Oracle are labeled with a CX Sales badge.



Metric Cards and Contents Containers

Each tab in your dashboard is composed of two elements that hold the tab content:

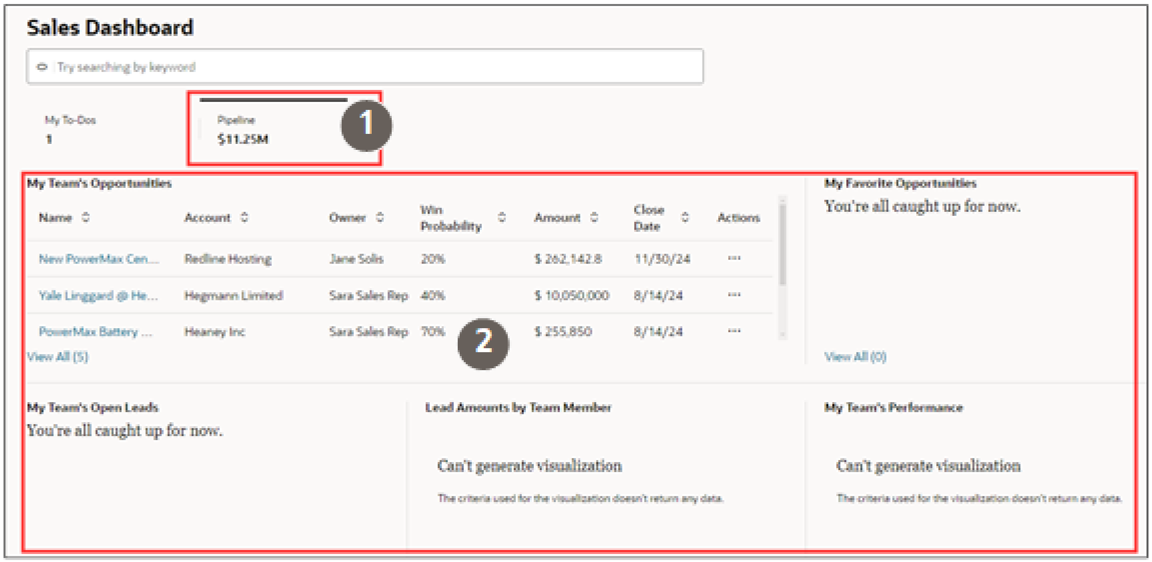
- **Metrics Card** (identified by callout 1 in the following screenshot)

  A metric card holds the information at the top of the tab, which is always visible on the dashboard even when the tab isn't selected. It contains the tab title, and other optional information. The predefined Pipeline metric card for sales managers, for example, calculates and displays the sum of opportunity revenue.

- **Contents Container** (callout 2)

  The contents container holds the information in the tab. For each tab, you can add up to 5 sections with tables, visualizations, and lists.

  The predefined pipeline contents container, for example, displays a table of opportunities in the first section. Other sections display My Appointments, My Team's Open Tasks, My Open Tasks, and My Team's Overdue Tasks. All present information from different saved searches.



To the user, the metrics container and the contents container form a single unit. However, the metrics container and the contents container are separate entities. You must specify what information you want to display in each one separately, and you must reference the metrics container in the contents container to make both appear together when users click the tab.

Key VB Studio Components Overview

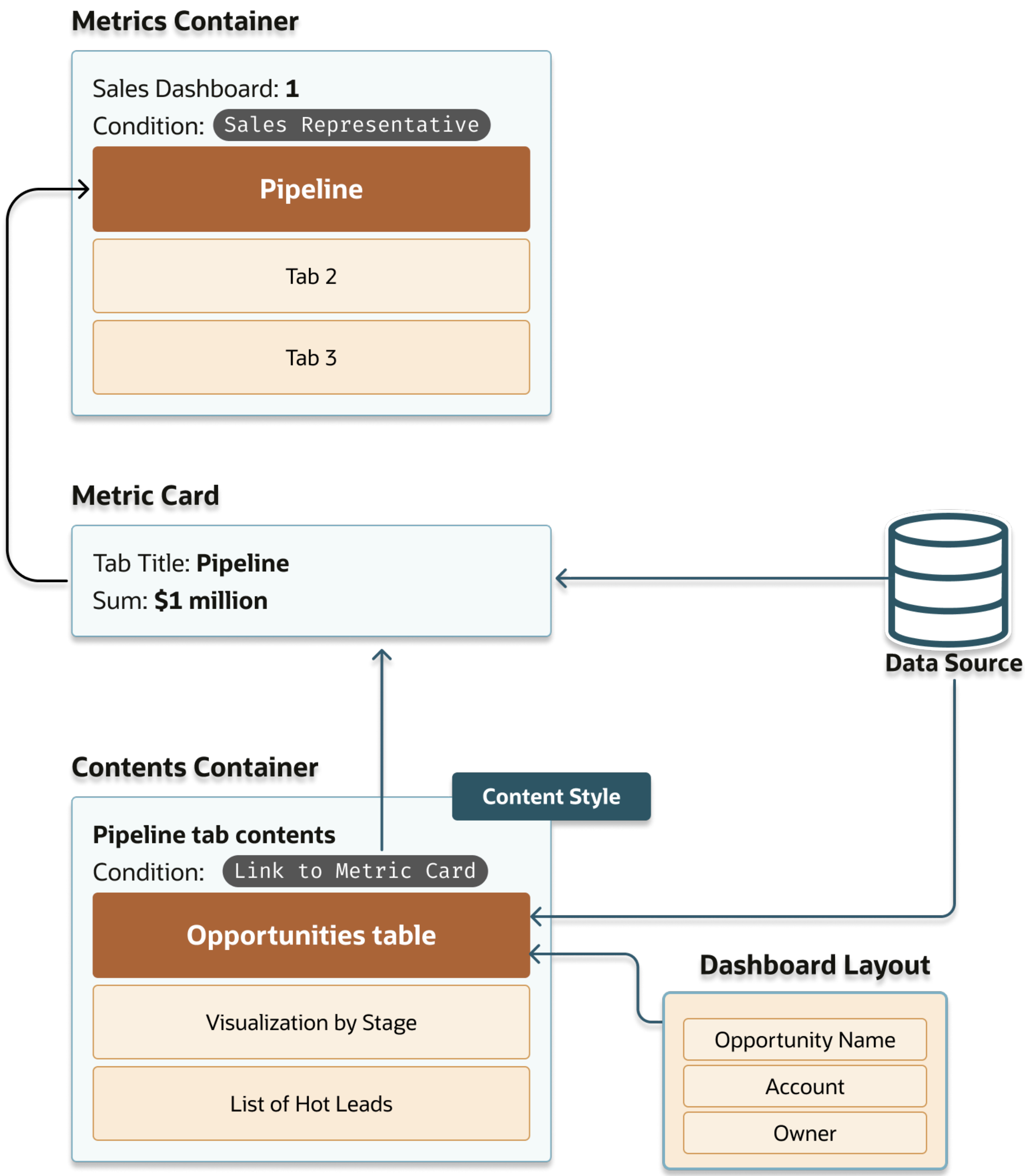
Here's an overview of the main VB Studio components that you use to create a custom dashboard. This example focuses on creating a tab displaying a table of opportunity pipeline information from a saved search.

- You create your dashboard as an instance of a **Metrics Container**.
- You add the dashboard tabs, called **Metric Cards**.

  The Sales Dashboard 1 metrics container in the diagram includes the different metric cards for each tab in the dashboard, including the Pipeline tab.
- In the code for each metric card, you reference the data source. This can be a saved search or an Oracle Transactional Business Information (OTBI) analysis. You enter the tab title, and you can calculate a key indicator such as the sum of opportunity revenue. You can also add badges and filters.
- You create the contents of your tab as a **Contents Container** that's linked to the metric card by its condition. You can add as many as 5 sections to each contents container. A section can hold a table, a visualization, or a list. You must reference the source of your data in each section.

  The source of your data can be a saved search, an OTBI analysis, an Express Report, or a visualization. The data source need not be the same as the source of data for the metric card.
- What attributes appear in a table or a list, in which order, and at what relative widths, is controlled by dynamic table layouts: **Dashboard Layout** for tables, and **List Layout** for lists. Each layout includes the common fields for each object, but you can substitute your own.

  The table layout doesn't apply to tables (Tabular Charts) created from visualizations.
- As you add sections, the layout of the tab itself changes according to the **Content Style**.


Metrics Container
Sales Dashboard: 1
Condition: Sales Representative
Pipeline
Tab 2
Tab 3
Metric Card
Tab Title: Pipeline
Sum: $1 million
Data Source
Contents Container
Content Style
Pipeline tab contents
Condition: Link to Metric Card
Opportunities table
Visualization by Stage
List of Hot Leads
Dashboard Layout
Opportunity Name
Account
Owner

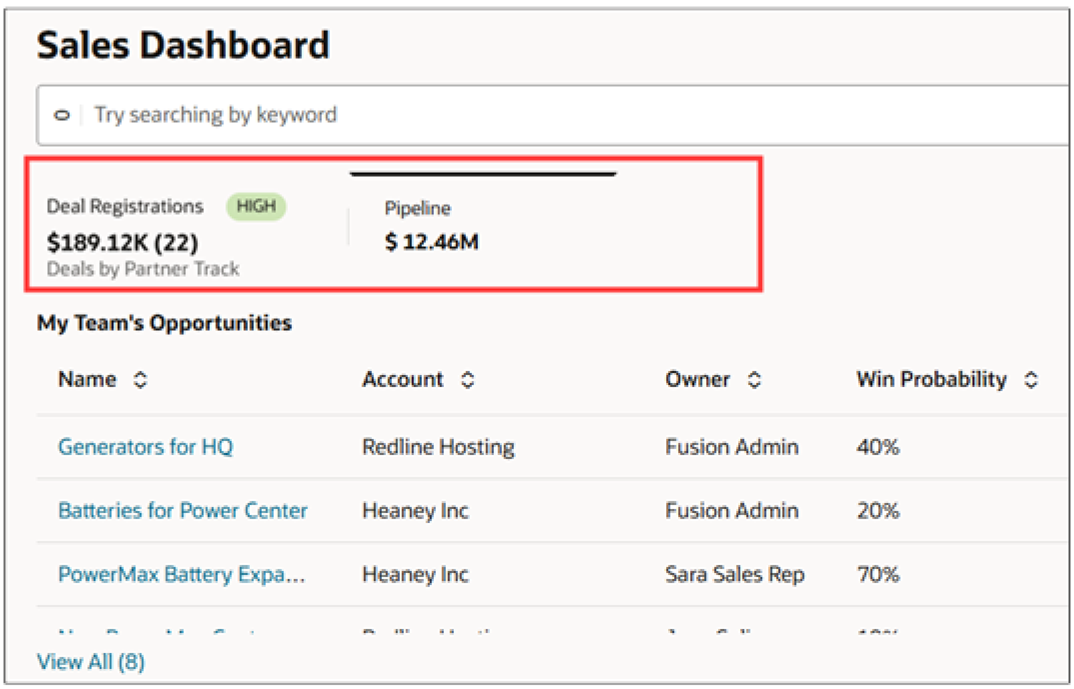
What Sources of Sales Information You Can Display in a Sales Dashboard and How

The Sales Dashboard can display tables, lists, and charts. The sales data you display can come from saved searches, Oracle Transactional Business Intelligence (OTBI) analyses, and Express Reports. You can also embed reports from Oracle Fusion Data Intelligence.

Information Summaries on Tabs

At the top of each tab, you can display important information, such as the sum of the opportunity revenue or the number of overdue tasks. Your data for the summary can come from Adaptive Search saved searches or OTBI analyses, but the calculation is done in the Metrics Card you create. You can create the contents of the Metrics Card as a visualization first to save some time.

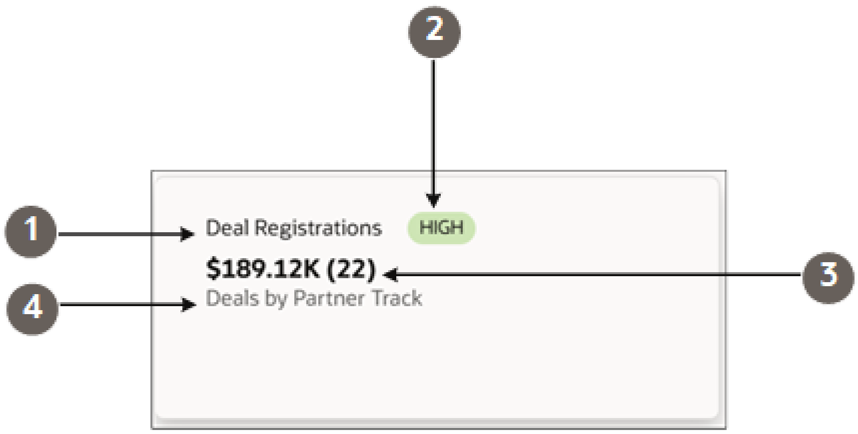
Here's a screenshot of a sample dashboard with 2 tabs. The screenshot highlights the part the tabs defined by metric cards.



Metric cards can have up to 5 different elements show in the screenshot of a sample partner deal registrations metric card:

Callout	Field	Description
1	Name	The title of your dashboard tab.
2	Badge	Displays badges you specify for different value ranges.

Callout	Field	Description
3	Primary Metric (Secondary Metric)	You can display up to 2 metrics. In this example, it's the sum of the Deal Size (Primary Metric) and the count (Secondary Metric).
4	Description	One-line description that you enter or that's filled in dynamically.

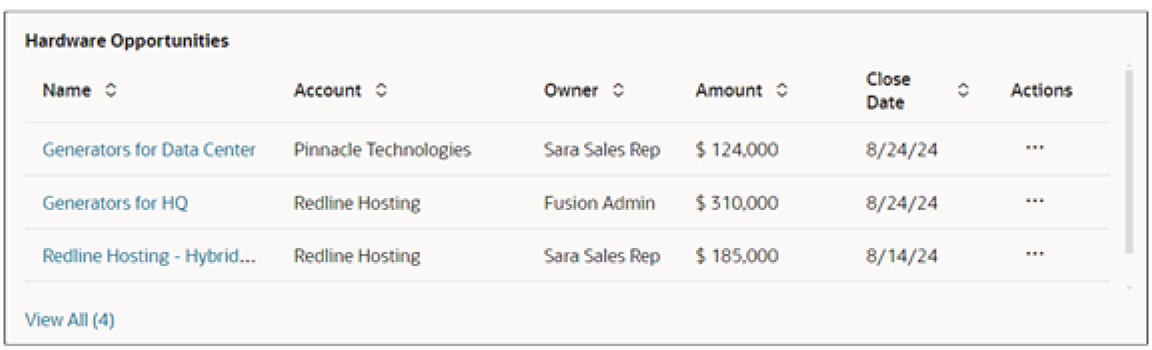


Tables

You can display information from Adaptive Search saved searches and OTBI analyses as table.

You can specify which columns permit drill-downs and specify different destinations for columns, if required. Each table includes a View All link that displays a page with up to 2000 records.

To display tables, you create a Tabular Chart visualization using the Visualization Configuration tool and then display that visualization. For saved searches, you can also create a table directly in the Visual Builder Studio.



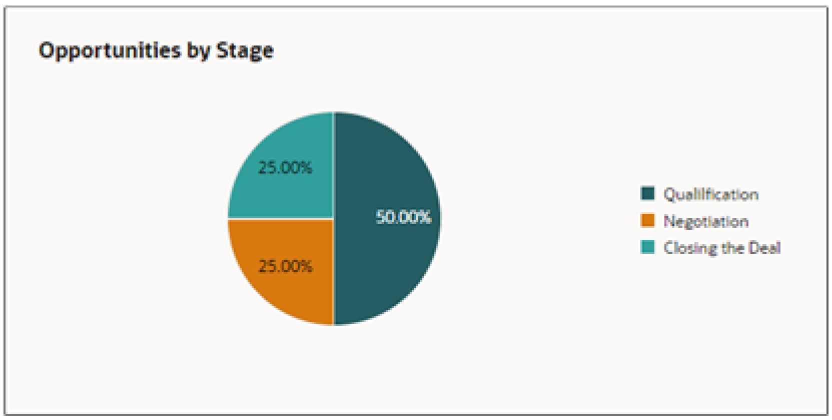
Lists

You can display saved searches as a list.

Charts

You can display charts that you created using the Visualization Configuration tool and charts from Express Reports. You can add charts from Express Report directly. To include charts from saved searches and OTBI analyses, you create visualizations using the Visualization Configuration tool first.

Salespeople can drill down on each section of the visualization to display the records it contains.



Embedded Oracle Fusion Data Intelligence Components

You can also embed reports from Oracle Fusion Data Intelligence. Fusion Data Intelligence is built on top of Oracle Analytics Cloud and includes the same subject areas you find in Oracle Transactional Business Intelligence.

How You Can Display Different Types of Data Sources

Here's a table summarizing how you can display sales data from different sources:

Sales Data Source	Tab Summary (Metric Card)	Table	List	Visualization
Saved Search	Yes	Yes	Yes	Yes
OTBI Analysis	Yes	Yes	No	Yes
Express Report	No	No	No	Yes

- **Saved Searches**

  Saved searches are the most versatile source of data for your dashboard:

  - You can add key performance indicators (KPIs) calculated from the saved searches to the metric card at the top of each tab.
  - You can display saved searches as a table or a list.
  - You can also create different types of charts from saved searches using the Visualization Configuration tool. Charts can link to the underlying data. A salesperson clicking on a slice of a pie chart, for example, sees the list of opportunities just for that slice.
- **OTBI Analyses**
  - You can add key performance indicators (KPIs) calculated from the analyses to the metric card at the top of each tab.
  - You can display Oracle Transactional Business Intelligence (OTBI) analyses in the tab either as a chart or a table by creating a visualization using the Visualization Configuration tool and then displaying that visualization.

- **Express Reports**

  You can add an express report to the dashboard, but only the chart is displayed, not the accompanying table.

Overview of Sales Dashboard Setup Steps

Use this overview to guide you through the initial setup of sales dashboards. You must create your own sales dashboards regardless of the types of edits you want to make. Even a small wording change requires you to create your own. You can't edit any of the sales dashboard content provided by Oracle directly.

Step	Description	More Details
1	If you're using Oracle Transactional Business Intelligence (OTBI), then you can take advantage of the predefined analyses provided by Oracle. All you need to do is to unarchive them in OTBI and they're ready for creating the charts and tables in the next step.	*Unarchive Analyses Provided by Oracle*
2	Create the charts and tables that you want to display in the sales dashboard using the Visualization Configuration tool that's available in Application Composer. You can create tables and charts from Adaptive Search saved searches and from OTBI analyses.	*Visualizations*
3	Create the shell of your sales dashboard by duplicating one of the 2 dashboards provided by Oracle. In VB Studio, the dashboards are represented as 2 instances of the metrics container. Who can view a sales dashboard is determined by the condition and the position of the metrics container instance in the hierarchy of cases. To ensure you can view the copy, clear the conditions to make the dashboard visible to everyone.	*Create the Shell of Your Dashboard by Duplicating On of the Dashboards Provided by Oracle*
4	Within the dashboard shell container, create the metric card for each tab you're adding. The metric card is the top of the tab that's always visible.	*Create the Metric Card for the Top of the Tab*
5	Create the shell for your tab content by duplicating one of the contents containers provided by Oracle. To make the content display when users click the tab, you must link your content container to the metric card you created by editing the condition.	*Create the Contents Container and Link It to the Metri Card*
6	In each tab, you can have up to 5 sections displaying different sales information as tables, lists, and charts. Before you start adding content to the contents container for your tab, review how the layout style controls the display of the sections in the tab. The size of your section in the tab depends on the number of sections you add and the section's position on the tab.	*How Information Displays in a Sales Dashboard Tab a You Add Sections*
7	Adding sections predefined by Oracle is as easy as clicking **Add** and selecting the content.	*Add Predefined Content to Your Tab*

Step	Description	More Details
8	Adding your own content involves creating new sections and entering parameters in code. **Note:** Although the UI permits it, you can't copy and edit the code predefined by Oracle.	*Add the Sections to the Tab's Contents Container*
9	The parameters you enter are different for displaying a table, a list, a visualization, and for embedding a report from Oracle Fusion Data Intelligence. The values of some parameters differ by business object. The sample code provided in the topics is specific to opportunities. For other business objects, substitute the values listed in the *Values to Use in Code for Standard Components in Tables, Metric Cards, and List Views* topic.	To learn about the code and parameters, see these topics: • *Display a Saved Search as a Table If You Didn't Create It As a Visualization* • *Display a Saved Search as a List* • *Add a Custom Visualization to Your Dashboard Tab* • *Add My List* • *Embed Content Created in Oracle Fusion Data Intelligence*
10	For adding tables and lists, a layout specifies which attributes appear as columns, in which order, and at which relative widths. The layouts, which are specific to each business object, already include key information, but you can add attributes or change their order and relative width.	*Layouts That Specify What Fields Display in Tables and Lists and How* *Specify the Columns in a Sales Dashboard Table and Their Widths*
11	Expose filters that salespeople can use to narrow down the scope of the sales information they view on the sales dashboard tabs. By applying filters, they can narrow down the information to a specified period of time or select specific team members.	*Enable Filtering of Sales Dashboard Information*
12	Specify which of the sales dashboard tabs you want to display by default when users open the sales dashboard.	*Specify which Dashboard Tab Displays by Default*
13	Enable sales dashboard personalization.	*Enable Sales Dashboard Personalization*
14	You can preview your changes. If you're satisfied, you can publish your changes to Git and merge them to the main code line.	*Preview Your Changes and Save Them to the Git Repository*

2 Before You Begin

Before You Configure the Sales Dashboard

Before your team can start creating application extensions, you must first set up Oracle Visual Builder Studio. You only need to set up VB Studio once for every implementation.

Complete VB Studio implementation steps are documented in the *Oracle Cloud Administering Visual Builder Studio* guide. See the topic: *How Do I Set Up VB Studio?*

Required: Set the Extension ID for Sales

When using VB Studio to extend Sales pages, your extension must use the extension ID: `site_cxsales_Extension`. You set this extension ID when you first set up your project.

A project collects all the people, tools, and processes you need to complete a discrete software effort in VB Studio. Oracle best practice dictates that you use a single project for all the extension work you do within the Oracle Cloud Application environment family.

You can create this project using one of two methods discussed in the following video: *Create the Visual Builder Studio Project*.

Each method requires a different way to set the extension ID:

- Create a project from a Sales page by clicking the **Edit Page in Visual Builder Studio** link in the **Settings and Actions** menu. This is the recommended method to create a project because it automates the creation of key VB Studio components. See the topic: *Create a Simple Extension*.

  If you choose this method, then you'll update your project's extension ID to `site_cxsales_Extension` by editing the extension-level settings. See the topic: *Establish Extension-Level Settings*.
- Create a project from the Organization home page. See the topic: *Manually Create a Project for Extensions*.

  If you choose this method, then you'll enter the required extension ID when you create your own workspace. See the topic: *Create an Extension*.

Note: Be sure to publish your extension so that the updated extension ID becomes the default going forward for everyone else working on the extension.

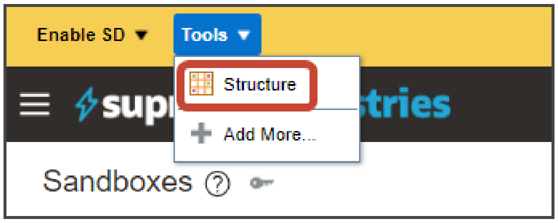
Tip: Create Additional Workspaces

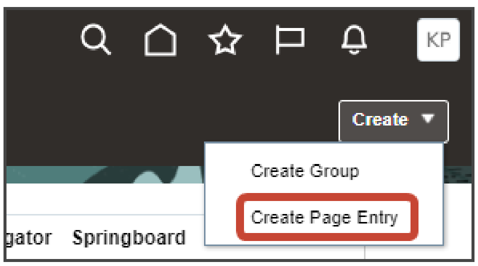
At some point in your extension lifecycle, you might need to create a new workspace in an existing project. You may want to create a new workspace from the main branch if you forget what changes a particular workspace contains, for example. Follow the instructions in the topic: *Clone an Existing Repository*.

You can also view the following video: *Create a Workspace*.

Enable Navigation to the Sales Dashboard

Enable the navigation to the Sales Dashboard from the Home page.

1. From the Navigator, select **Configuration** > **Sandboxes**.
2. On the Sandboxes page, click **Create Sandbox**.
3. On the Create Sandbox page, enter a sandbox name.
4. For the Publishable option, click **Yes**.
5. In the All Tools table, select the **Structure** checkbox.
6. Click **Create and Enter**.
7. On the Sandboxes page, from the **Tools** menu, select **Structure**.
   Here's an example image showing how to access the Structure tool:
   

8. On the Navigation Configuration page, click **Create** > **Create Page Entry**.
   Here's a sample image of the create a page entry item in the UI:
   

9. On the Create Page Entry page, enter these field values:

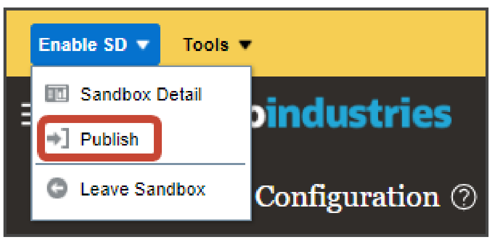
Create Page Entry Field Values for Sales Dashboard

Field	Value
Name	Sales Dashboard
Icon	Select your desired icon
Group	(Top Level)
Show on Navigator	Yes

Field	Value
Show on Springboard	Yes
Mobile Enabled	Yes
Link Type	Dynamic URL
Web Application	ORA_FSCM_UI
Destination for Web Application	/redwood/cx-sales/application/container/dashboards/sales-dashboard

10. Click **Save and Close**.
11. Test and confirm that you can access the Sales Dashboard using both the Navigator and from the Home page.
12. When ready, publish the sandbox by clicking the sandbox name and then clicking **Publish**.

    This sample image shows the Publish item when publishing a sandbox:



Enable Opportunity Pipeline Attributes in Adaptive Search

For the Sales Dashboard to display values in the Opportunity Pipeline amount, Win Probability, and Amount fields, a few fields must be enabled in Oracle Sales Adaptive Search. These fields aren't enabled by default. After you enable the fields, you must run the Partial Publish scheduled process to activate your changes.

Enable the following fields in the **Setup** > **Advanced** tab of the **Configure Adaptive Search** setup task:

- Primary Revenue

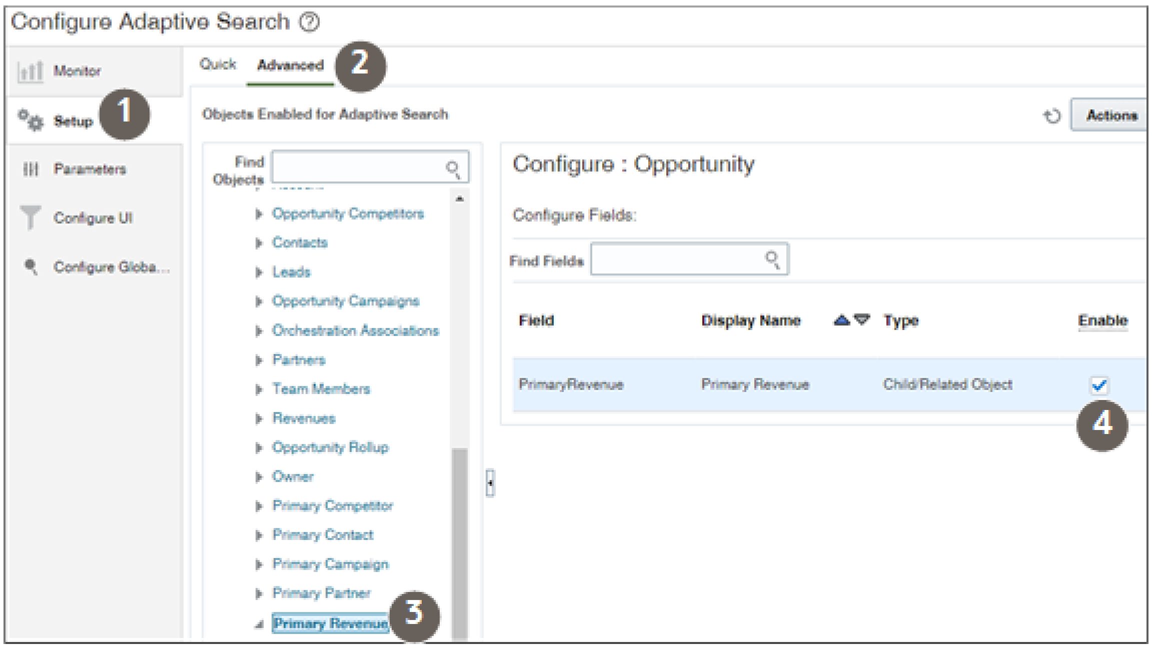
  Navigation: **Opportunity** > **Primary Revenue** > **Primary Revenue**
- Win Probability

  Navigation **Opportunity** > **Primary Revenue** > **Opportunity Revenue** > **Primary Revenue** > **Win Probability**
- Amount

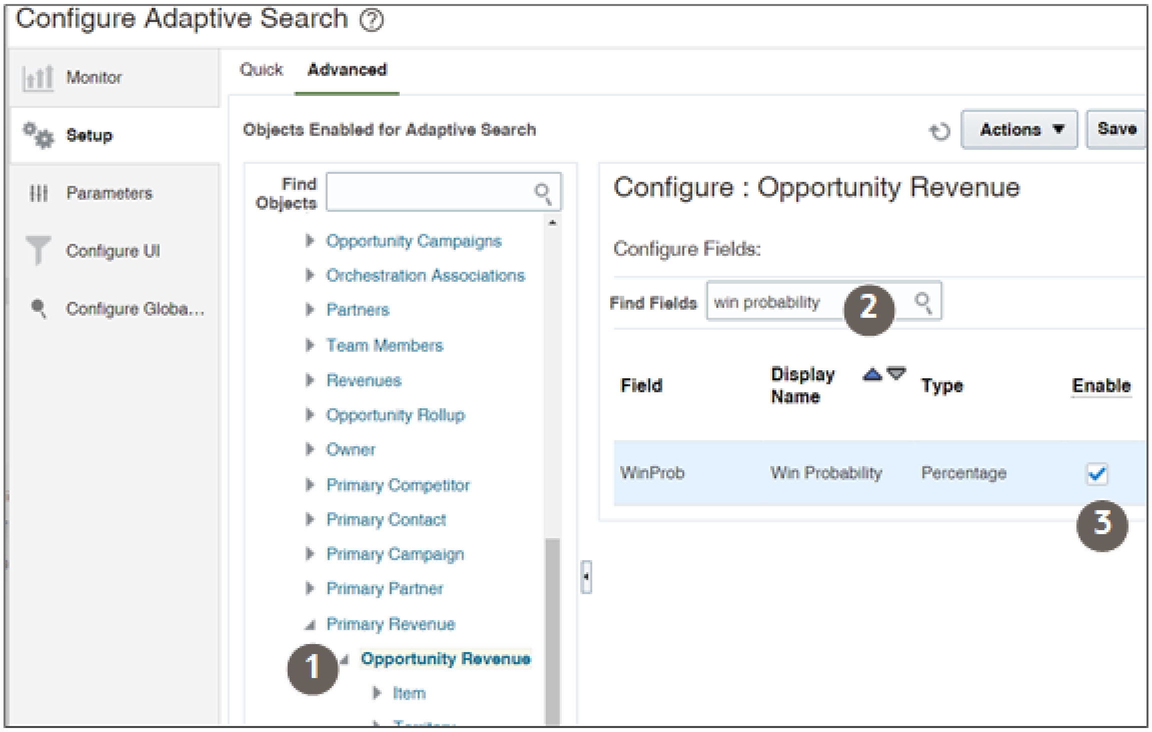
  Navigation: **Opportunity** > **Primary Revenue** > **Opportunity Revenue** > **Amount**

Here's how to enable the fields:

1. In the Setup and Maintenance work area, open the **Configure Adaptive Search** task:
   - Offering: Sales
   - Functional Area: Sales Foundation
   - Show: All Tasks
   - Task: Configure Adaptive Search
2. Click **Setup** > **Advanced**.
3. On the Advanced tab, click **Primary Revenue**.
4. In the Configure: Primary Revenue pane, select the **Enable** checkbox for the **Primary Revenue** field.



5. Click **Primary Revenue** > **Opportunity Revenue**.



6. In the Configure: Opportunity Revenue pane, use the **Find Fields** field to search and enable the **Amount** field and the **Win Probability** field.
7. Click **Save**.
8. Publish the new attributes by running the Partial Publish process:
   a. Click **Actions** > **Partial Publish**.
   b. Select **Opportunity**.
   c. Click **Proceed with Partial Publish**.
   d. In the Partial Publish window, click **Publish**.

Unarchive Analyses Provided by Oracle

Oracle provides you with sample Oracle Transactional Business Intelligence (OTBI) analyses for accounts, activities, forecasts, leads, opportunities, orchestrations, and quotes. You can edit them to suit your needs. Here's how to unarchive them into your OTBI library.

1. Open Reports and Analytics (**Navigator** > **Tools** > **Reports and Analytics**).
2. Click **Browse Catalog**.
3. Open the **Shared Folder Custom** folder.
4. Locate and select **Oracle Sales Dashboard Content** folder.
5. In the Tasks pane, click **Unarchive**.

3 Create Charts and Tables as Visualizations

Visualizations

Using the Visualization Configurations tool in Application Composer, you can create interactive visual representations (called visualizations) of your sales data. The visualizations that you can create include bar charts, pie charts, combo charts, and funnel charts. You can also create tables (called Tabular Charts). You can add visualizations to the tabs in sales dashboards or to the CX Sales Mobile app. The Visualization Configuration tool also simplifies the creation of metric cards (the tops of the tabs in the sales dashboard).

You can create visualizations from two types of sources of sales data: from Adaptive Search saved searches and from Oracle Transactional Business Intelligence (OTBI) analyses. Each type requires slightly different setup so the setup steps are covered in different sections:

- For Adaptive Search saved searches, see the section *Create Charts and Tables from Saved Searches*.
- For OTBI analyses, see the section *Create Charts and Tables from OTBI Analyses*.

Prerequisites

To create visualizations, you must have the privilege ZCA_VIEW_DATA_VISUALIZATION_CONFIGURATION_PRIV.

Setup users with one of the following job roles provided by Oracle already have this privilege:

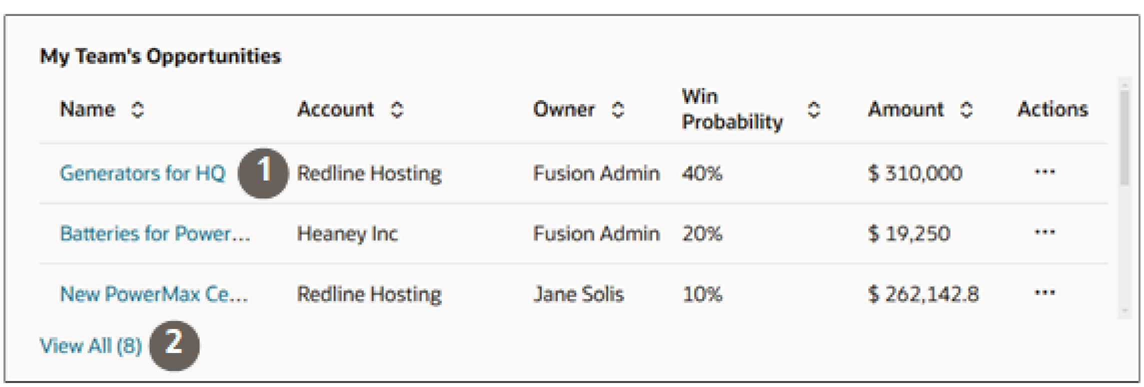
- ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB (Customer Relationship Management Administrator)
- ORA_ZBS_SALES_ADMINISTRATOR_JOB (Sales Administrator)

Table and Chart Visualizations at Runtime

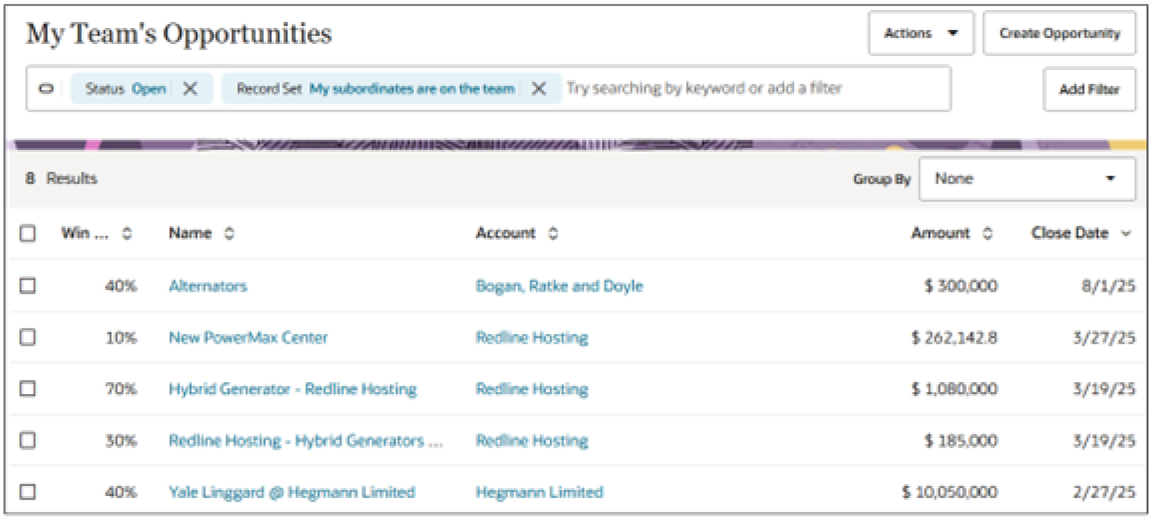
Here's how tables and charts created as visualizations display to users.

Table Visualizations

A table visualization displays the most frequently accessed records in a table. You can enable drill-downs to individual records and you can add the Actions list. With the smart actions, salespeople can create tasks, schedule appointments, send emails, and make calls directly from the table, for example.



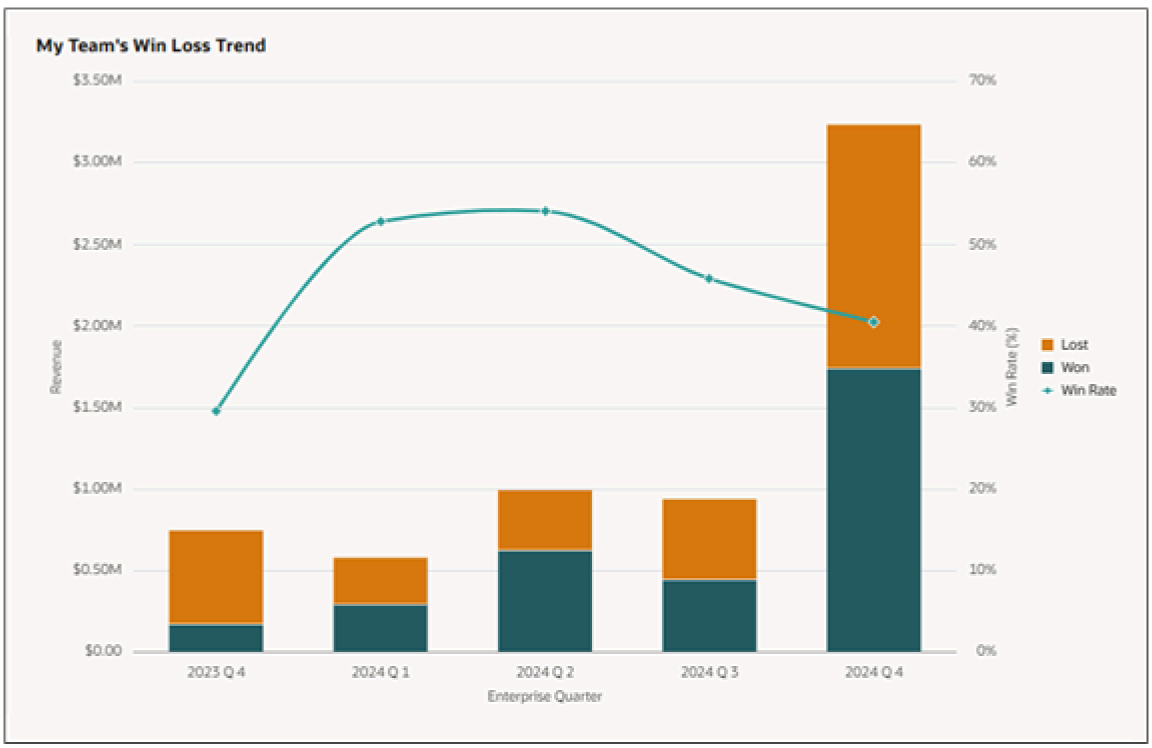
Tables include a **View All** link (callout 2 in the screenshot) that opens a list page showing all records matching the visualization selection criteria.



Win ...	Name	Account	Amount	Close Date
40%	Alternators	Bogan, Ratke and Doyle	$ 300,000	8/1/25
10%	New PowerMax Center	Redline Hosting	$ 262,142.8	3/27/25
70%	Hybrid Generator - Redline Hosting	Redline Hosting	$ 1,080,000	3/19/25
30%	Redline Hosting - Hybrid Generators ...	Redline Hosting	$ 185,000	3/19/25
40%	Yale Linggard @ Hegmann Limited	Hegmann Limited	$ 10,050,000	2/27/25

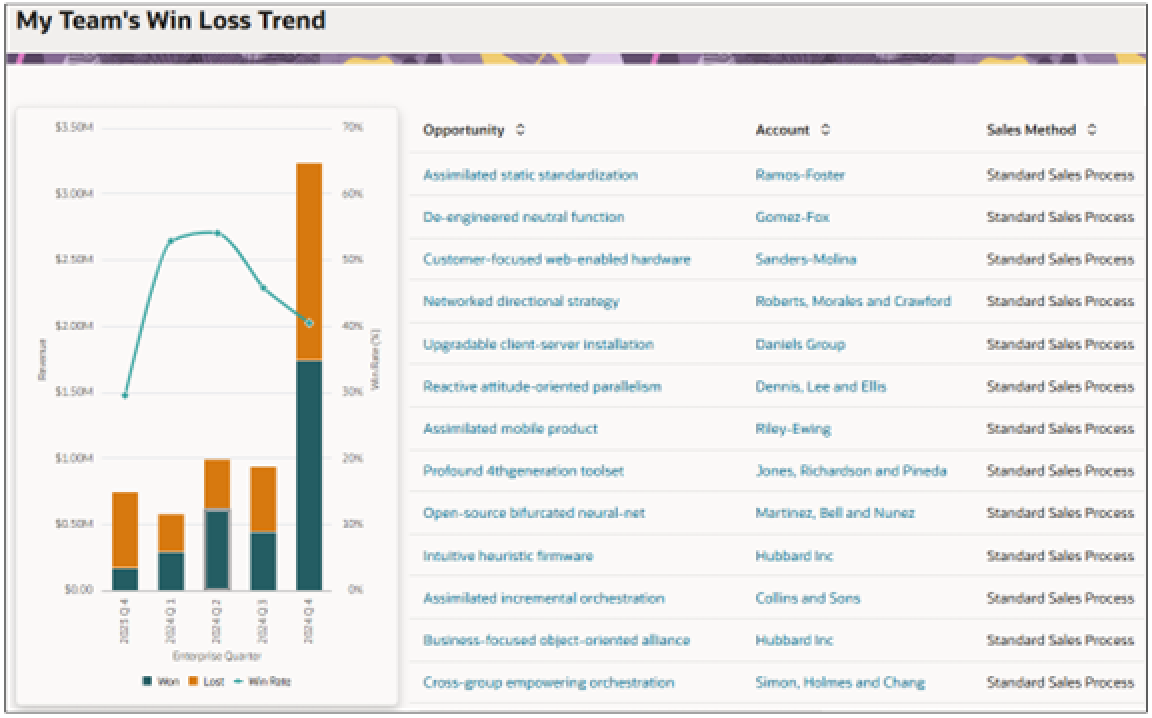
Chart Visualizations

Salespeople can click a section of a chart to view the underlying records.

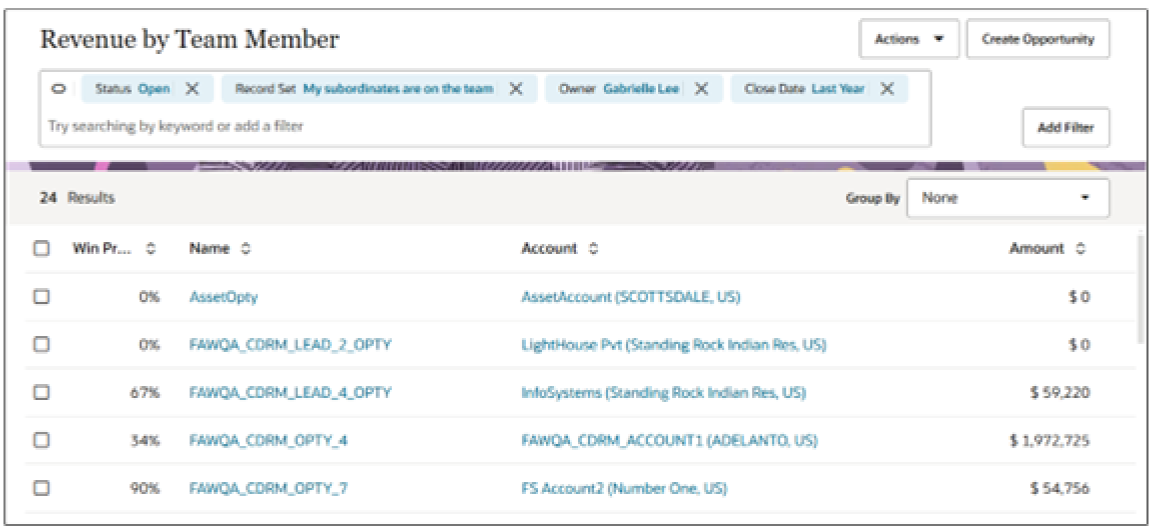


In the drill-down page, the list of records in that section displays in a side panel or a list page.

OTBI analyses open on a page showing both the visualization and the list.

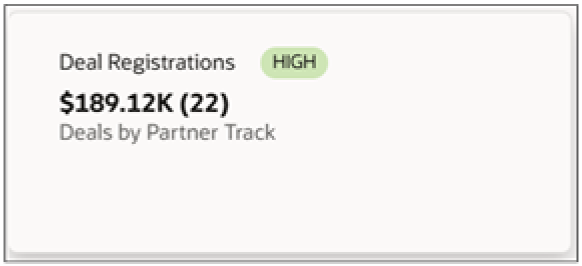


Adaptive Search saved search visualizations open a list page.



Metric Cards

You can also use the Visualization Configuration feature to speed up the creation of the information that appears at the top of each tab in the sales dashboard. Metric Card visualizations provide a UI where you can specify all of the information you can display, including the title and badges.



Because Metric Cards are used for configuring the sales dashboard tabs, they're covered in the Configure the Redwood Sales Dashboard chapter topic *Create the Metric Card the Quick Way: As a Visualization Configuration*.

Create Charts and Tables from Saved Searches

There are two ways you can create tables and charts from saved searches using the Visualization Configuration feature in Application Composer:

- You select the default saved search as the data source and enter the filters you need for the table or chart directly in the Create Visualization page.

  Entering filters in the visualization configuration makes it unnecessary to create and manage an individual saved search for each visualization.

  To learn how, see the topics:
  - *Create a Chart from the Default Saved Search*
  - *Create a Table from the Default Saved Search*
- You can also select a specific saved search and inherit the filters and the columns displayed in the list page from that saved search.

  To learn how, see the topics:
  - *Create a Chart from a Specific Saved Search*
  - *Create a Table from a Specific Saved Search*

Create a Chart from the Default Saved Search

You can save time when you select **Default** as your saved search in the Create Configuration page. That's because selecting the default saved search as your source, makes it possible for you to change the filters directly in the Visualization Configuration tool. Entering filters directly, eliminates the need for you to create and manage a specific saved search for each visualization.

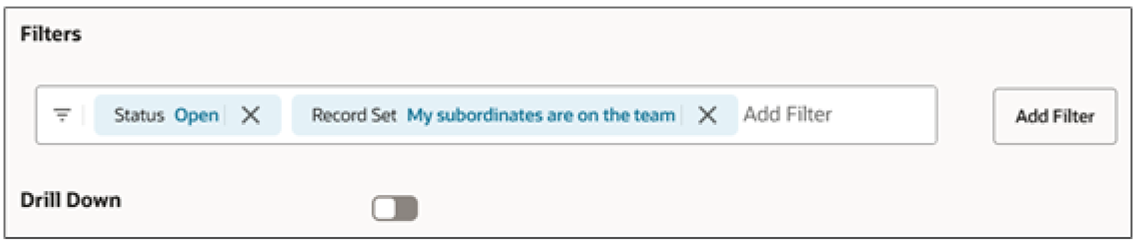
The visualization you create is based on the default saved search for the object that's shared with the sales organization as a whole.

1. Navigate to **Configuration** > **Application Composer**.
2. Click **Visualization Configuration** in the right-hand pane.
3. In the Visualization Configurations page, click **Add**.
4. From the Create Configuration page, **Source Type** field, select the **Adaptive Search**.
5. Select the object.

6. Leave the **Saved Search** field set to **Default**. This selection specifies to use the default saved search for the object.
7. In the **Visualization Type** field, choose the visualization type, such as a bar chart, pie chart, donut chart, combo chart and so on.
8. In the **Details** section, enter the dimensions and other preferences for the visualization. Which attributes are available depends on the type of visualization you're creating. The values you can enter in the attributes depend on your Adaptive Search setup.

   > **Note:** Only attributes that have been enabled for Group By in Adaptive Search are available for selection as your chart axes. For example, if you're creating a chart of opportunities by team member, then you must enable the Group By option for both Team Members and Team Member attributes in Adaptive Search setup. For details, see the topic *How do enable the filtering of visualizations by team members?*

   .
9. The **Filter** bar initially displays the filters in the default saved search. You can update or delete these filters and your own filters.

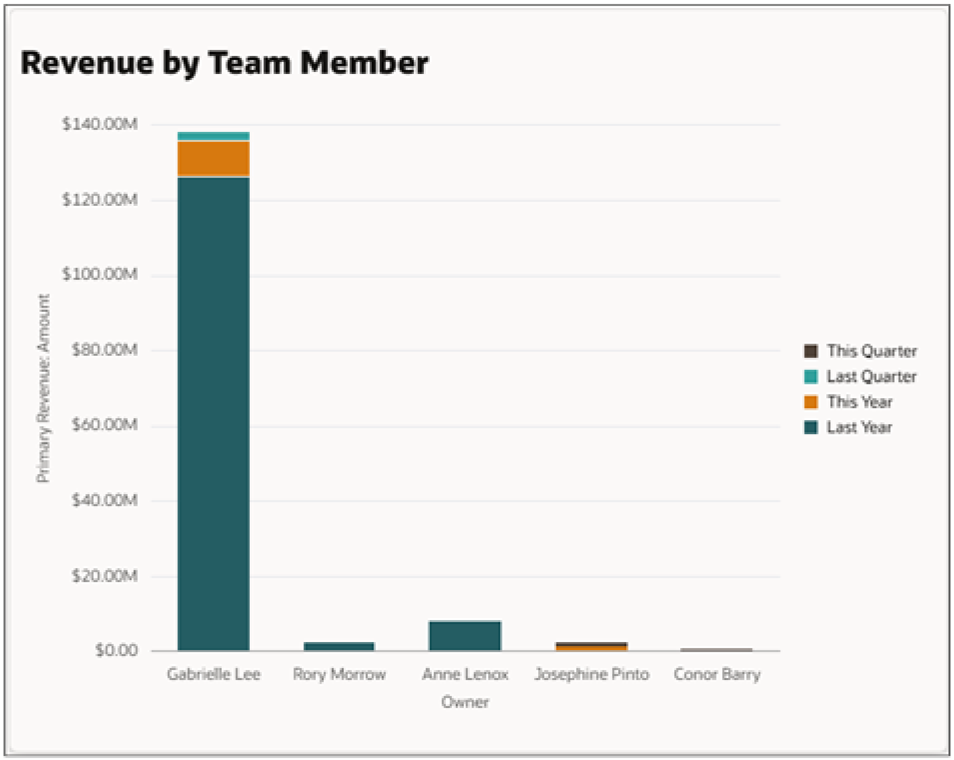


10. Enable salespeople to drill down to the list page by turning on **Drill Down**.

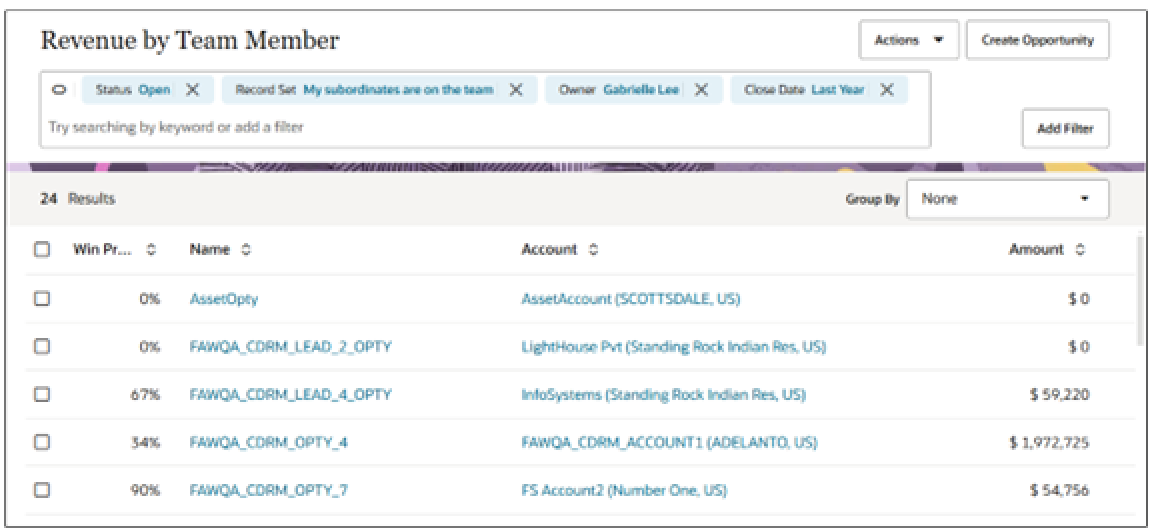
    When salespeople click one of the sections of the chart, the application displays a drill down page with a list of the records in that section. The columns are provided by the default saved search.
11. Set the **Status** field to **Active**.
12. Click **Create** to save the configuration.
13. Note the number of the visualization configuration on the list page. You'll use the number to include the visualization in the Sales Dashboard.

Entries for a Sample Revenue Bar Chart Visualization Created from Adaptive Search

Here are the entries to create a sample bar chart visualization from an Adaptive Search. The chart shows the revenue by opportunity team member for the last year and this year.



Clicking on last year's section for one of the team members opens a list page with the team member and last year's filters added. The list page includes the title of the visualization.



Here are sample entries to create the chart:

Field Name	Entry	Description
Source Type	Adaptive Search	Visualization source.
Object	Opportunities	Business object

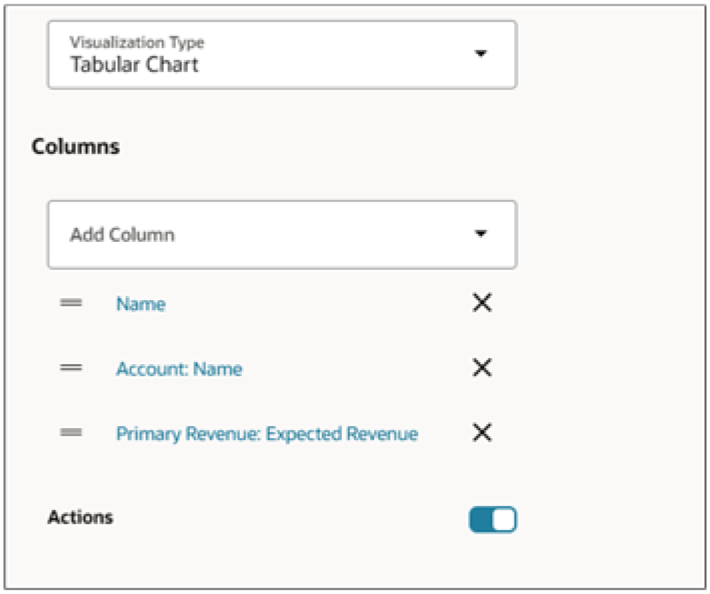
Field Name	Entry	Description
Saved Search	Default	The visualization uses the default saved search as the source of your information and lets you change the filters.
Name	Revenue by Team Member	The title of your visualization and the drill down page.
Visualization Type	Bar Chart	Different visualizations require different attributes.
X Axis	Owner	Displays owner names along the x-axis
Categorized By	Close Date	The quarter in which the close date appears. Because Stacked Layout is selected, the quarters appear as sections of each owner's bar.
Value	Primary Revenue Amount	Opportunity revenue
Format	Currency	Supplies currency symbol. Click the Preferences button to see this option.
Unit	Round Even	Rounding.
Aggregation	Sum	Type of calculation.
Orientation	Vertical	Bar chart orientation.
Stacked Layout	On	Displays the different quarters as sections on a bar. If off, you get separate bars for each time period for each team member.
Filters	Status: Open, Record Set: My subordinates are on the team	Filters come from the default saved search, but you can change them.
Drill Down	On	Enables drill-down.

Create a Table from the Default Saved Search

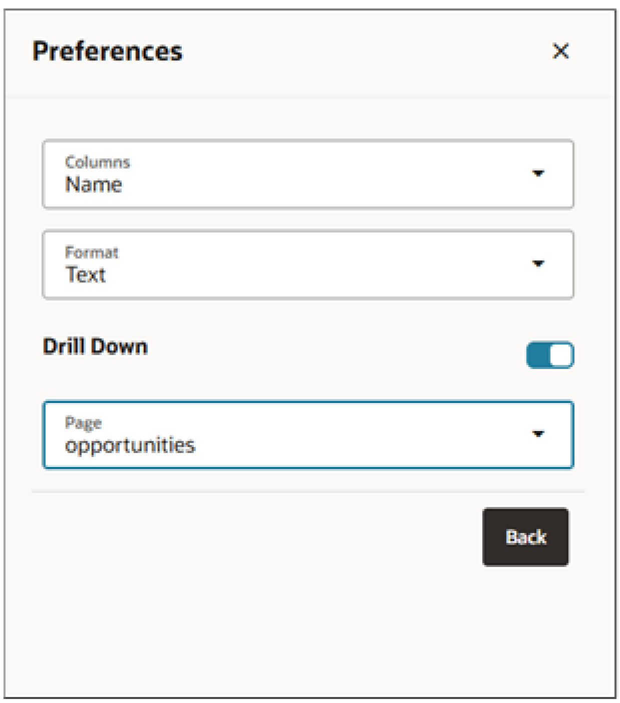
Here's how to create a table when you select Default as your saved search. Making this selection makes it possible for you to add the filters in the visualization itself.

1. Navigate to **Configuration** > **Application Composer**.
2. Click **Visualization Configuration** in the right-hand pane.
3. In the Visualization Configurations page, click **Add**.
4. From the Create Configuration page, **Source Type** field, select the **Adaptive Search**.
5. Select the object.
6. Leave the **Saved Search** field set to **Default**.
7. In the **Visualization Type** field, select **Tabular Chart**.

8. In the **Columns** section, use the **Add Column** list to add the columns you'd like to display in the table. The table builds as you add columns.

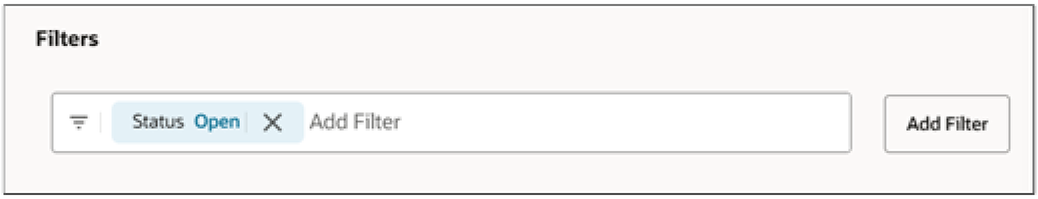


9. Turn on **Actions** to display the **Actions** list (three dot icon) on each record in the table.
10. Click the column name link to format the columns in the table and to specify the drill down destination:
    a. In the Preferences drawer **Format** field, add any required format for the column. For example, to add the currency symbol, select **Currency**.
    b. Click **Drill Down** to enable drill down from the column.
    c. Select the page you want to open. For example, to open the opportunity details page, select **opportunities**.

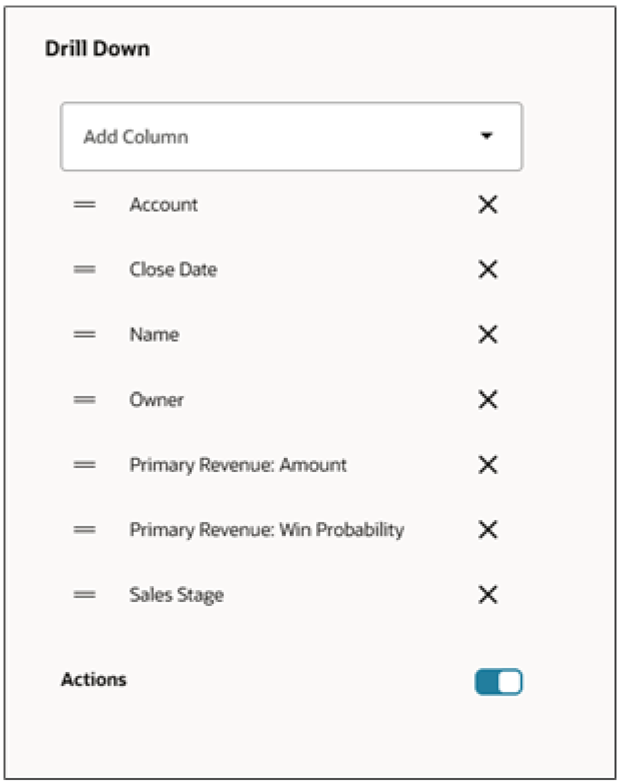
    d. Click **Back**.

11. In the **Filter** field, specify the filters for the table. The field initially displays the filters in the default saved search, but you can update, delete, and add filters.



12. In the **Drill Down** section, use the controls to specify which columns you want to display in the list page that opens when users click the **View All** link at the bottom of the table.

    Initially, the application lists the columns provided by the default saved search.
13. You can display the Actions menu on each record in the list by switching on **Actions**.

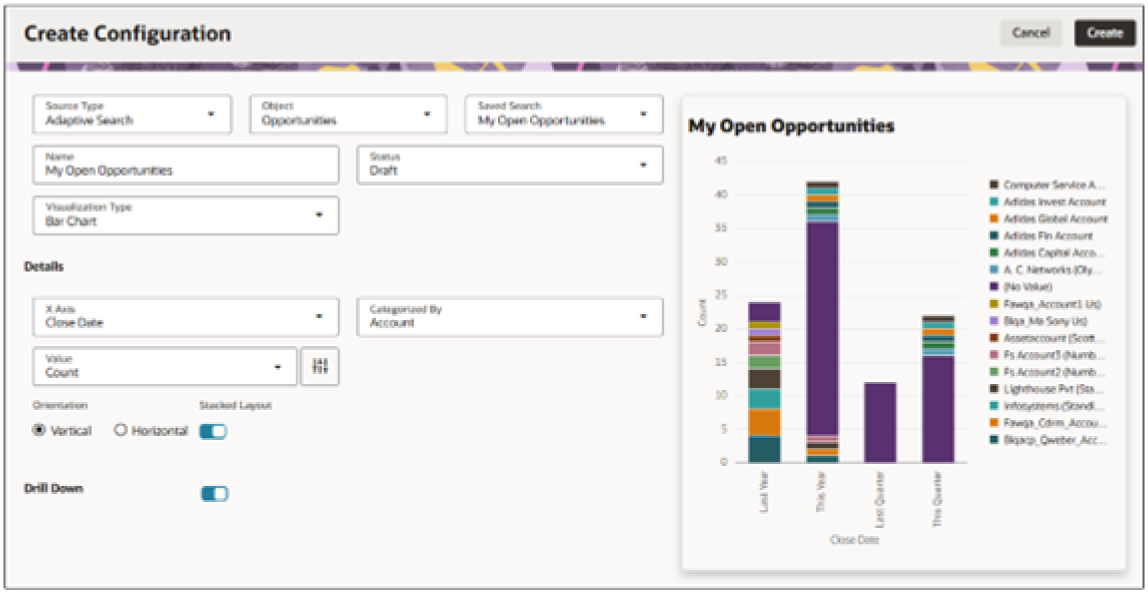


14. Set the **Status** field to **Active**.
15. Click **Create** to save the configuration.
16. Note the number of the visualization configuration on the list page. You'll use the number to include the visualization in the Sales Dashboard.

Create a Chart from a Specific Saved Search

You can create a table or a chart visualization from a specific saved search. This visualization type inherits the filters and the columns specified in the saved search. When users drill down on a section of the chart, they're directed to the page showing the saved search in the list page with the additional filter. When they click the View All link they see the list page for the saved search.

1. Make sure that the data in the saved search is visible to the user performing this configuration. You can temporarily remove filters from saved searches to ensure you see data.
2. Navigate to **Configuration** > **Application Composer**.
3. Click **Visualization Configuration** in the right-hand pane.
4. In the Visualization Configurations page, click **Add**.
5. From the Create Configuration page, **Source Type** field, select the **Adaptive Search**.



6. Select the object.
7. In the **Saved Search** field, select the saved search.
8. In the **Visualization Type** field, choose your preferred visualization, such as a bar chart, pie chart, donut chart, and so on.
9. In the **Details** section, enter the dimensions and other preferences for the visualization.

> **Note:** Only attributes that have been enabled for Group By in Adaptive Search are available for selection as your chart axes. For example, if you're creating a chart of opportunities by team member, then you must enable the Group By option for both Team Members and Team Member attributes in Adaptive Search setup. For details, see the topic *How do enable the filtering of visualizations by team members?*

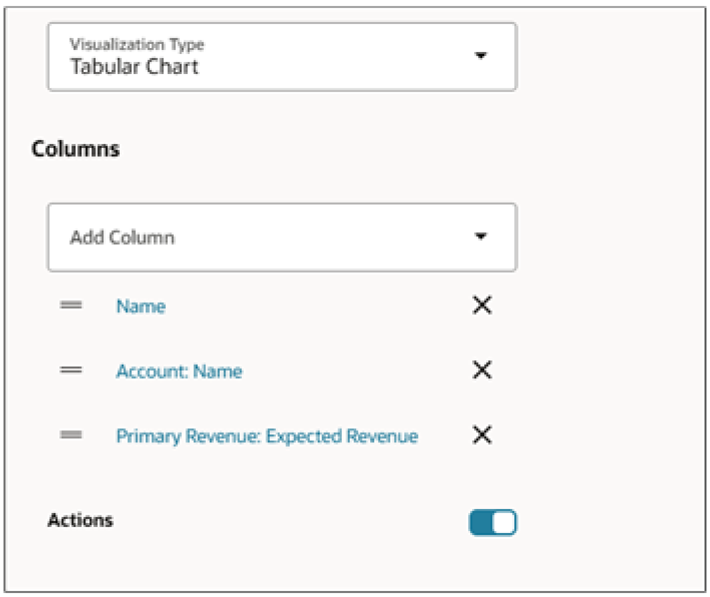
10. Enable salespeople to drill down to the list page from sections of the chart. Turn on **Drill Down**.
11. Set the **Status** field to **Active**.
12. Click **Create** to save the configuration.
13. Note the number of the visualization configuration on the list page. You'll need it when you add the visualization to the tab contents.

Create a Table from a Specific Saved Search

Here's how to create a table from a specific saved search.

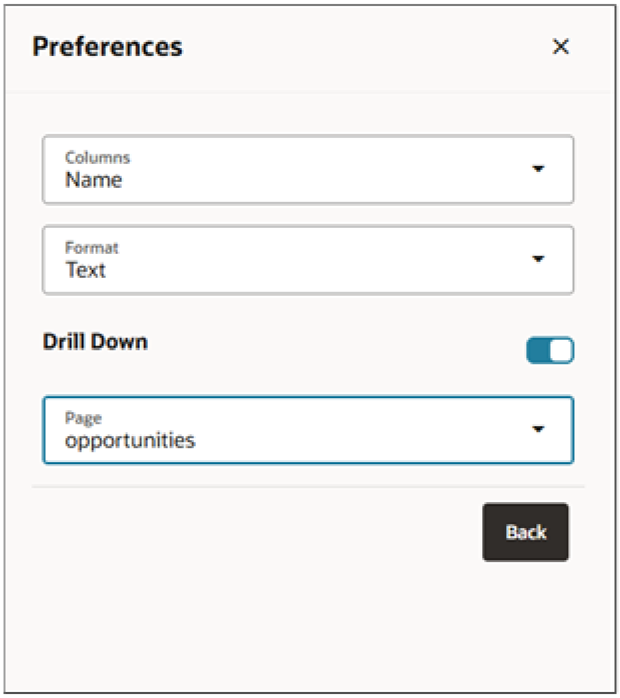
Note: The steps are very much the same as the ones you use to create a table from the default saved search. What's different here is that you can't change the filters and you can't choose which columns users see on the list page when they click the View All link at the bottom of the table. Both the filters and the list page column configuration come from the saved search.

1. Navigate to **Configuration** > **Application Composer**.
2. Click **Visualization Configuration** in the right-hand pane.
3. In the Visualization Configurations page, click **Add**.
4. From the Create Configuration page, **Source Type** field, select the **Adaptive Search**.
5. Select the object.
6. In the **Saved Search** field, select the saved search.
7. In the **Visualization Type** field, select **Tabular Chart**.
8. In the **Columns** section, use the **Add Column** list to add the columns you'd like to display in the table. The table builds as you add columns.



9. Turn on **Actions** to display the **Actions** list (three dot icon) on each record in the table.

10. Click the column name link to format the columns in the table and to specify the drill down destination:
    a. In the Preferences drawer **Format** field, add any required format for the column. For example, to add the currency symbol, select **Currency**.
    b. Click **Drill Down** to enable drill down from the column.
    c. Select the page you want to open. For example, to open the opportunity details page, select **opportunities**.



    d. Click **Back**.
11. Set the **Status** field to **Active**.
12. Click **Create** to save the configuration.
13. Note the number of the visualization configuration on the list page. You'll use the number to include the visualization in the Sales Dashboard.

Create Charts and Tables from OTBI Analyses

Using the Visualization Configuration tool in Application Composer, you can create interactive charts and tables for your sales dashboard from Oracle Transactional Business Intelligence (OTBI) analyses. Salespeople remain in the Redwood User Experience when they drill down on charts or tables. They can drill down to individual records and take advantage of smart actions, such as making a call by clicking a phone number.

The steps to create a chart or a table use the same UI, but are slightly different, so they're covered in separate topics:

- *Create a Table from an OTBI Analysis*
- *Create a Chart from an OTBI Analysis*

What to Check in OTBI Before You Start

- Make sure that the person creating the visualization can view data in the analysis itself. You can temporarily remove filters from your analysis to ensure that you see data as you create the visualization.
- Make sure that your analysis includes only those columns that you display in the visualization.
- If you plan to enable drill-down from a table on the sales dashboard, then you must also add the ID and PUID columns that can uniquely identify each individual record to your analysis.

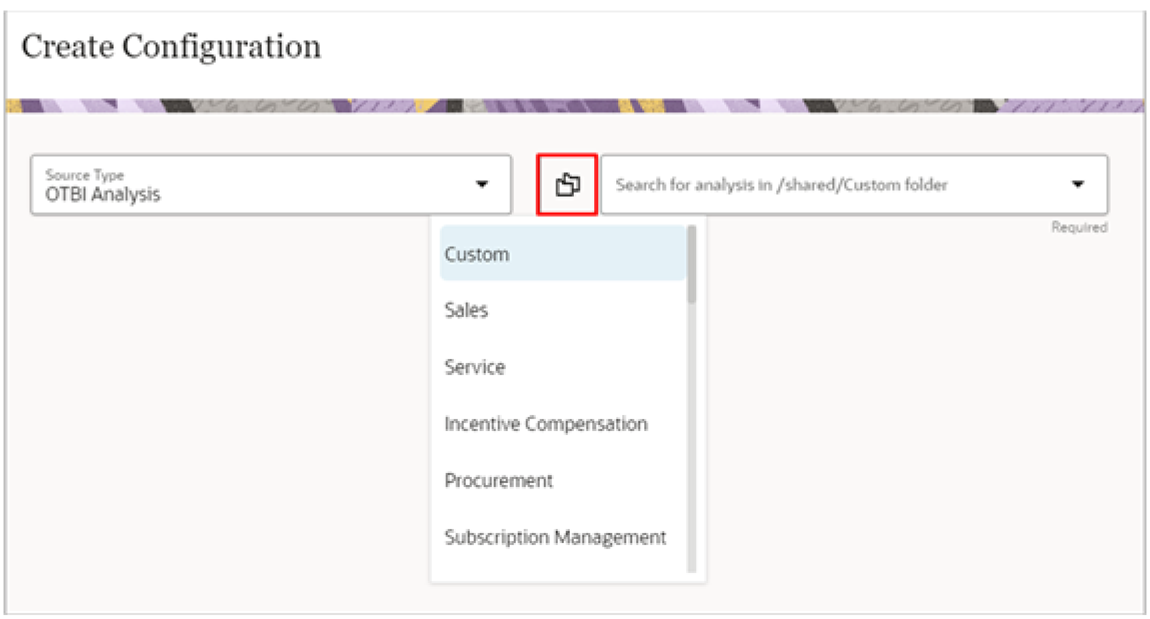
  For example, to enable drill down on opportunity name, you include the Opportunity Number (the PUID) and the Opportunity ID columns. For accounts, it's Account Number and Account ID. You get the idea.
- You must ensure that none of the columns you selected as criteria in your OTBI analysis are hidden in the Results tab. All of the columns, including the ID and PUID must be visible in the results.
- Your analysis must be stored in the Custom folder.

Create a Table from an OTBI Analysis

Here's how to create a table for the sales dashboard from an OTBI analysis using the Visualization Configuration feature in Application Composer.

Create the Visualization

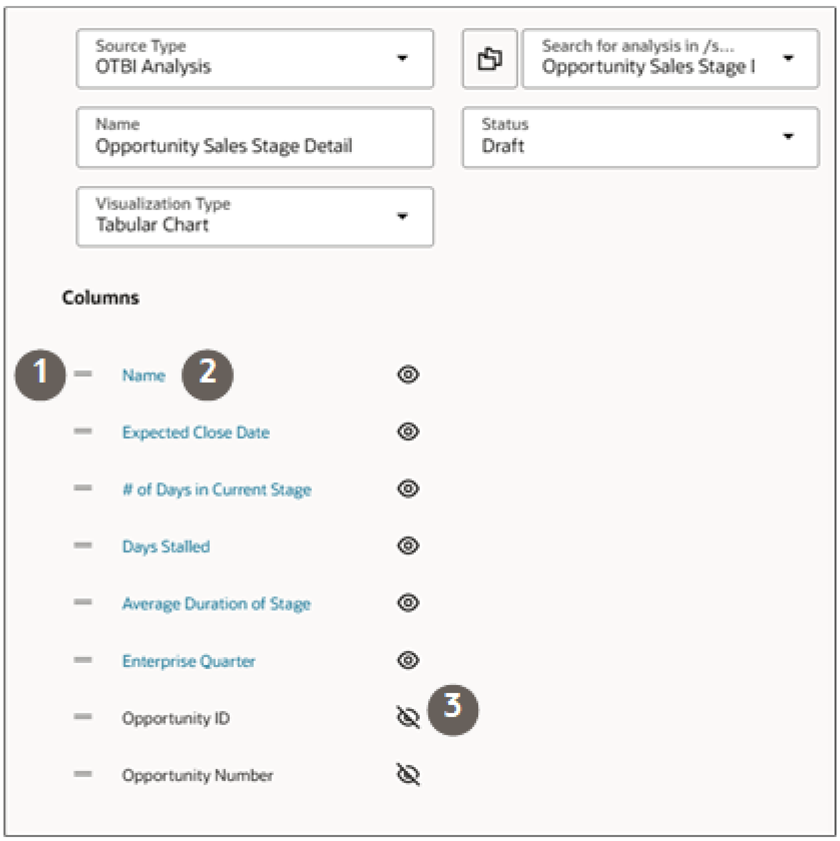
1. Navigate to **Configuration** > **Application Composer**.
2. Click **Visualization Configuration** in the right-hand pane.
3. Click **Add** to create a new configuration.
4. On the Create Configuration page, select **OTBI Analysis** as the **Source Type**.
5. Click **Select Catalog Folder**, the folder icon, and select **Custom**.



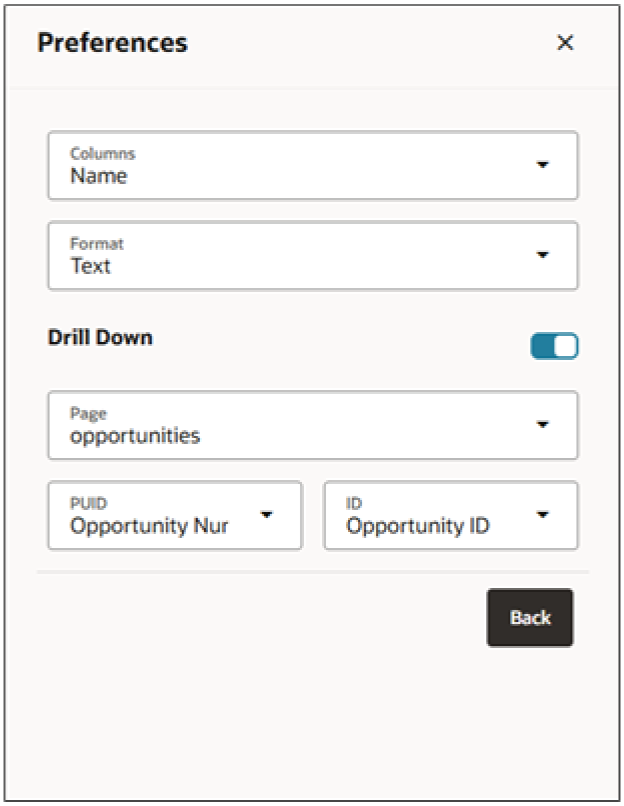
6. In the search field, enter the name of your custom analysis.
7. Select the analysis in the search results. All the fields in the analysis are included automatically, and the visualization displays on the page.

8. In the **Name** field, enter the title for the table.
9. In the **Visualization Type** field, select**Tabular Chart**.
10. Use the controls in the **Columns** section to change column order and to hide columns you don't want to display in the table.

Callout	Control Description
1	Use the handles to change column order.
2	Click the name link to specify column format (such as currency, decimal, or percentage) and enable drill-down on the field.
3	Hide columns in the table. In this example, we're hiding the **Opportunity ID** and **Opportunity Number** columns. These columns are used to identify the record for the drill down in the next step. They aren't meaningful to salespeople.



11. Enable drill-down functionality on fields in the table. For example, open an opportunity by clicking on its name:
    a. Click the **Name** link.
    b. On the Preferences drawer, turn on **Drill Down**.
    c. In the **Page** field, select the type of record you're opening. In this example, **opportunities**.
    d. In the **PUID** field, select the number ID for the object, for opportunities this is the Opportunity Number, for account's it's the Account Number, and so on.
    e. In the **ID** field, select the object ID field, such as Opportunity ID.



    f. Click **Back**.
12. You can test the drill-down from the table.
13. Set the configuration **Status** to **Active**.
14. Click **Create** to save the configuration.
15. Note the number of the visualization configuration on the list page. You use this number to reference the visualization in the sales dashboard.

Create a Chart from an OTBI Analysis

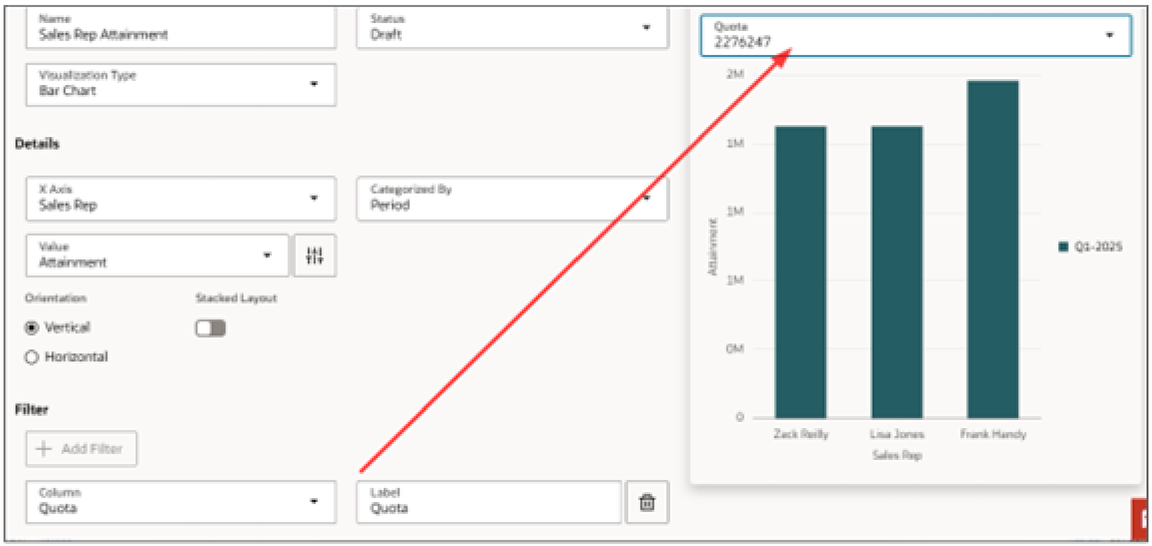
Follow these steps to create various kinds of charts from Oracle Transactional Business Intelligence (OTBI) analyses.

1. Make sure that the person creating the visualization can view data in OTBI itself. You can temporarily remove filters from your analysis to ensure you see data as you create the visualization.
2. Navigate to **Configuration** > **Application Composer**.
3. Click **Visualization Configuration** in the right-hand pane.
4. Click **Add** to create a new configuration.
5. On the Create Configuration page, select **OTBI Analysis** as the **Source Type**.

6. Click **Select Catalog Folder**, the folder icon, and select **Custom**.
7. In the search field, enter the name of your custom analysis.
8. Select the analysis from the search results. All the fields in the analysis are included automatically.

> **Note:** If the analysis you selected includes more columns than required, you receive a warning message that the visualization might not show the correct data. To resolve this issue, remove the columns that aren't used in the visualization. For example, a bar chart can support 2 dimensions, 1 metric, and you can have 1 filter attribute. If a report has more than 4 columns (3 dimensions and 1 fact), then you'll see this warning.

9. In the **Visualization Type** field, select the chart type.
10. In the **Name** field, enter the display title of the chart.
11. In the Details section, enter the parameters specific to the chart type you selected. Different types of visualization require you to enter different parameters.
12. Optionally, add a filter that salespeople can use to narrow down the scope of the visualization. The filter appears above the chart.



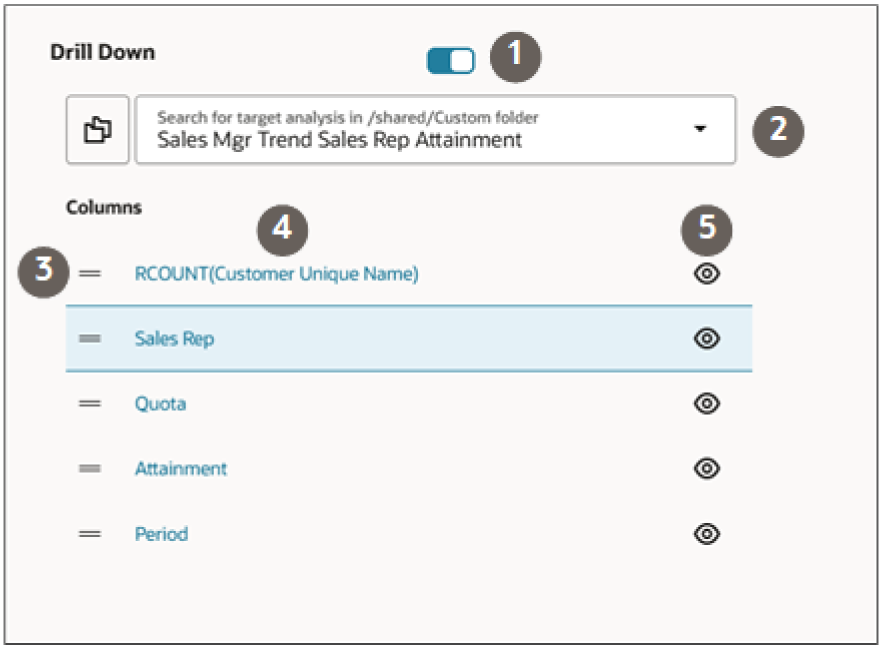
13. Enable drill-down functionality and configure the list page that salespeople see when they click a chart section.

    You can display the same or different analysis in the list page. For example, you can display an analysis with more details than the original.

    The target OTBI analysis must have a prompted filter applied on the same field. For example, if a manager clicks a bar of a chart representing an opportunity owner, then the analysis you take them to must include a filter for the resource.

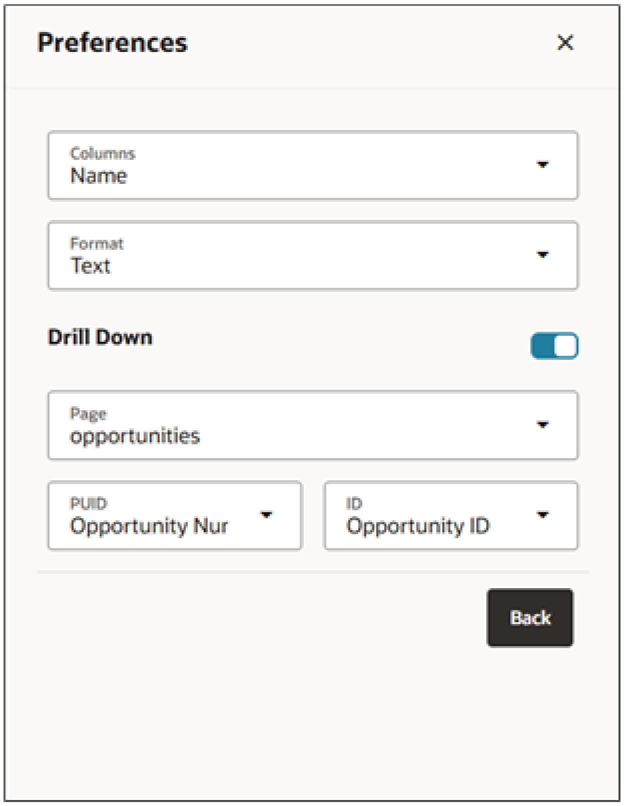
Callout	Field	Description
1	Drill Down switch	Enables users to click a section of the chart to view a page listing the records matching the filters in that section.
2	Search for target analysis	Select the analysis you want to open when users click a section of the chart.
3	Drag handle	Change the order of columns in the list page

Callout	Field	Description
4	Column names	Click to enable drill-down on the list page.
5	Hide	Hide columns on the list page



14. Enable drill-down on individual columns on the list page, specify their formatting, and the page they open:

    a. Click the field name link.
    b. On the Preferences drawer, turn on **Drill Down**.
    c. In the **Page** field, select the type of record you're opening. For example, **opportunities**.
    d. In the **PUID** field, select the number ID for the object, for opportunities this is the Opportunity Number, for accounts it's the Account Number, and so on.
    e. In the **ID** field, select the object ID, such as Opportunity ID.

    f. Click **Back**.
15. Set the **Status** field to **Active**.
16. Click **Create** to save the configuration.
17. Note the number of the visualization configuration on the list page.

Import and Export Visualization Configurations

You can import and export your visualization configurations using the Export Management tool.

How to Import and Export Visualization Configurations

To import and export visualization configurations, refer to the *Understanding Import and Export Management for Sales and Fusion Service* guide.

Object: Visualization Configuration

Sample scripts:

- To export a specific configuration: `ReportConfigNumber = 'CDRM_1002'`
- To export all configurations: `ReportConfigNumber IS NOT NULL`

How do enable the filtering of visualizations by team members?

To filter data by the different team members in saved searches and their visualizations, you must enable the Team Members attribute for the Group By feature in Adaptive Search. You can then use the Team Members field to group data in your search and the resulting visualization. Suppose, for example, that a bar chart in the Sales Dashboard shows performance for different team members. Enabling the attribute, makes it possible to click one of the bars to view the performance just for that team member.

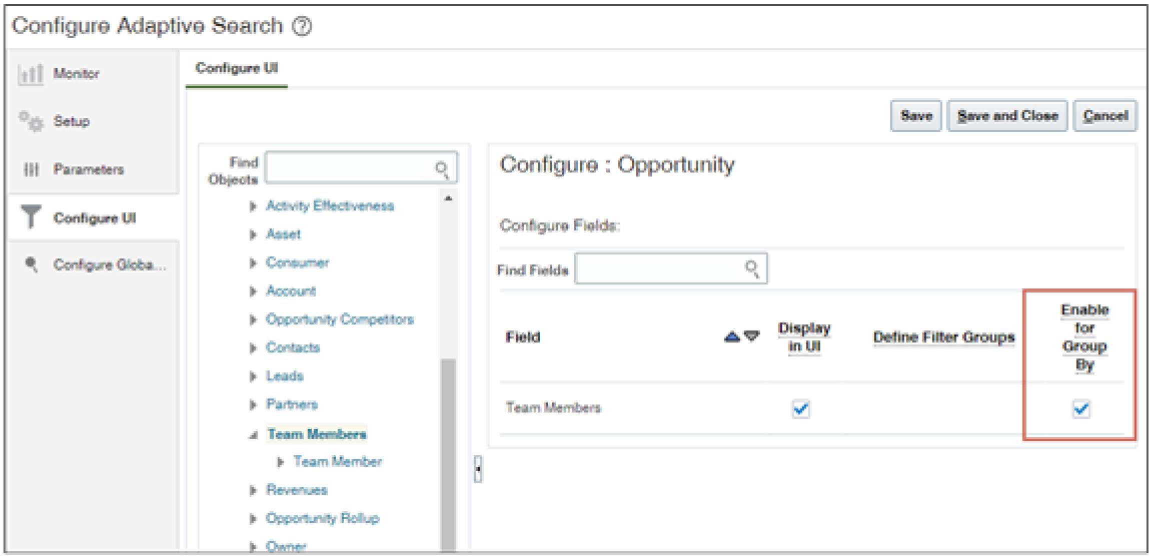
Enable the Group By Feature

1. Click **Navigator** > **My Enterprise** > **Setup and Maintenance**.
2. In the Setup and Maintenance work area, open the **Configure Adaptive Search** task.
    - Offering: Sales
    - Functional Area: Sales Foundation
    - Task: Configure Adaptive Search
3. On the **Configure Adaptive Search** page, click the **Parameters** tab.
4. Highlight the **Enable Group By** and click **Edit**.
5. In the Warning message, click **Yes**.
6. In the **Current Value** field, change the value from N to Y and click **Save and Close**.

Enable Team Member Fields for Group By

1. Sign in as a setup user.
2. Navigate to the Setup and Maintenance work area, and open the Configure Adaptive Search task:
    - Offering: Sales
    - Functional Area: Sales Foundation
    - Show: All Tasks
    - Task: Configure Adaptive Search
3. On the Configure Adaptive Search page, click the **Configure UI** tab.

4. Select the object you want to set up for your Group By in the left pane of the tab and open the object hierarchy to locate Team Members and Team Member.



5. Select the **Enable for Group By** option for both Team Members and Team Member fields.

> **Note:** Ensure that the **Display in UI** option is also selected

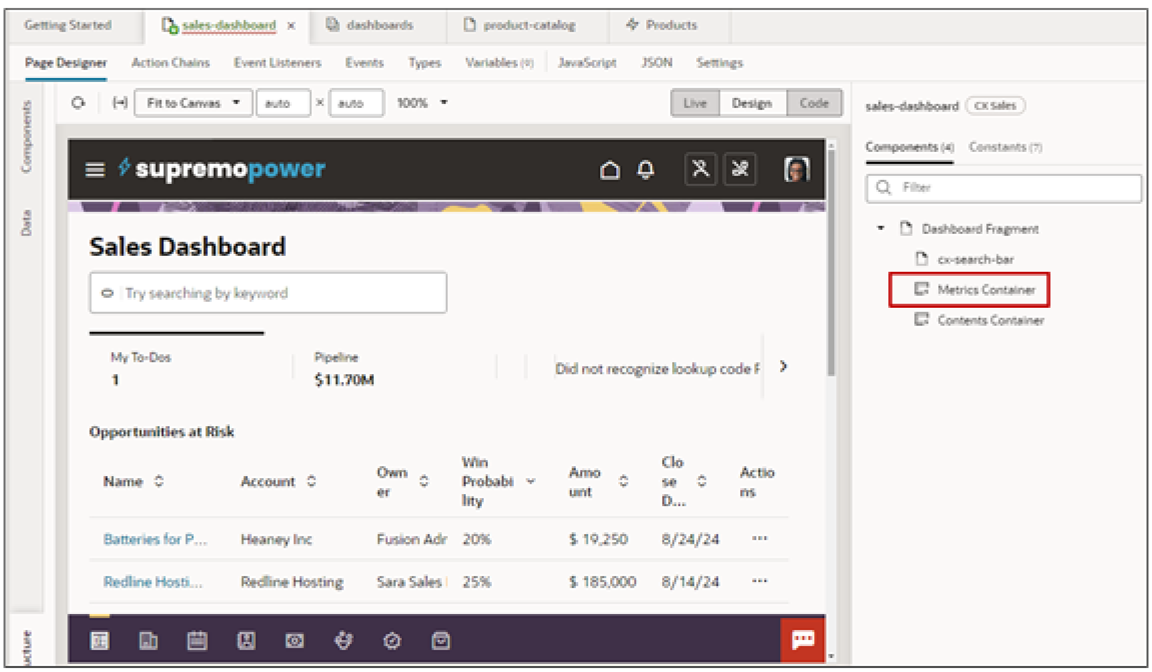
6. Click **Save and Close**.
7. Sign out and then sign in again to see your changes.

4 Configure the Dashboard

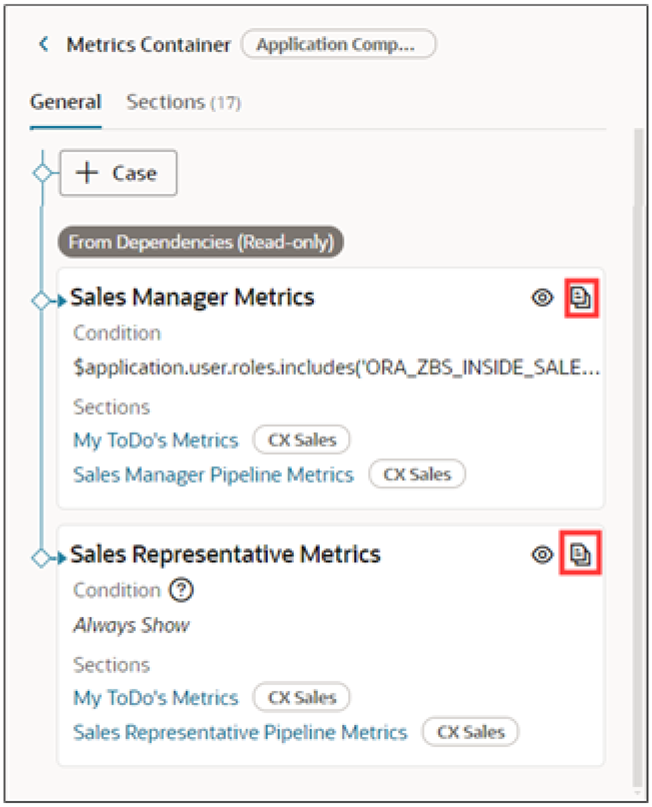
Create the Shell of Your Dashboard by Duplicating One of the Dashboards Provided by Oracle

Oracle provides you with 2 sales dashboards. One's tailored to sales representatives and the other to sales managers. The one for sales managers includes conditions that make it visible only the two sales manager roles provided by Oracle. The one for sales representatives includes no conditions, so it's available to everyone else. Before making changes or creating a new dashboard, make a copy. You can't edit the dashboards provided by Oracle, even if you're making a minor change.

1. Open the Sales Dashboard in the application and click **Settings and Actions** > **Edit Page in Visual Builder Studio**.
2. On the Components pane, click the **Metrics Container** link.

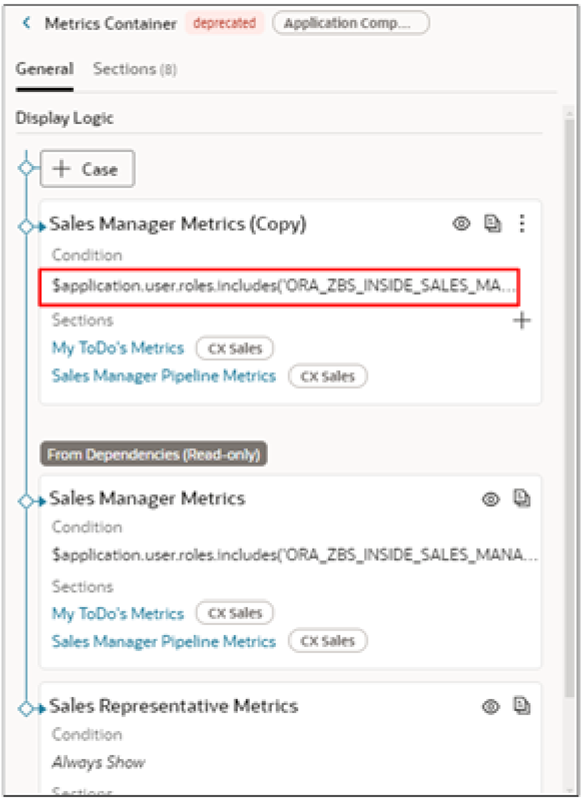


3. In the Components pane, duplicate either the **Sales Representative Metrics** or the **Sales Manager Metrics** containers by clicking **Duplicate**.



4. To enable you to view your sales dashboard during setup, edit the **Condition** field of the duplicate you made and delete the contents. Deleting the condition displays the dashboard to all users, regardless of the job role.

Before you publish your dashboard, you'll add the appropriate condition to display the tab to the correct audience.



You can tailor your sales dashboard to the audience you want by adding any number of conditions. For example:

- To show the dashboard to users with specific job roles, you can add a condition such as: `$application.user.roles.includes('ORA_ZBS_INSIDE_SALES_MANAGER_JOB') || $application.user.roles.includes('ORA_ZBS_SALES_MANAGER_JOB')`
- To have the dashboard appear for a single user, you can add a condition such as the following: `$user.userName=='SALES_ADMIN'`

5. The conditions you enter are evaluated in order from the top instance (case) down. So, make sure that the sales dashboard you're creating is in the correct position in the list of cases.

   For example, the Sales Manager Metrics dashboard provided by Oracle displays to sales manager roles because it's the first on the list. All other users see the Sales Representative Metrics dashboard.
6. You're now ready to start creating the metric cards (the tabs on the dashboard) as described in the topic *Create the Metric Card for the Top of the Tab*
7. You can also edit the existing dashboard contents. You can:

   - Delete any of the existing metric cards.
   - Change their position using the arrow icons. The position in list of Sections determines the order of the tabs in the sales dashboard.

Create the Metric Card for the Top of the Tab

For each tab you want to add or extend, you must create a custom metric card. You can't edit or duplicate any of the metric cards provided by Oracle.

There are two ways of creating a metric card:

- Create the metric card as a visualization using the Visualization Configuration tool in Application Composer.

  This method simplifies some of the steps, including the configuration of badges, that you'd have to complete in code.

  For details, see the topic *Create the Metric Card the Quick Way: As a Visualization Configuration*.
- Create the metric card directly in Oracle Visual Builder Studio.

  For details, see the topic *Create the Metric Card Directly in VB Studio*

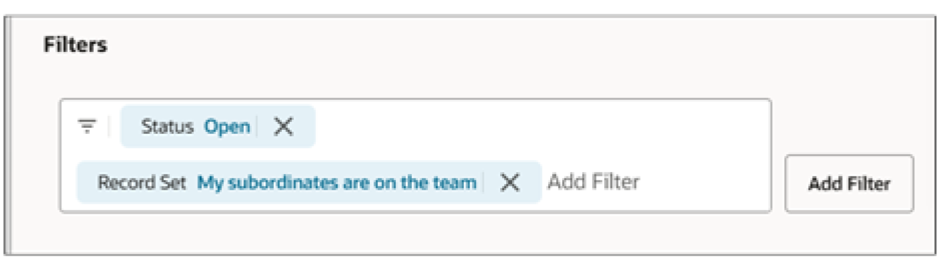
Create the Metric Card the Quick Way: As a Visualization Configuration

You can speed up the creation of a metric card when you create it as a visualization configuration. This method provides a UI for several technical steps, including the creation of badges.

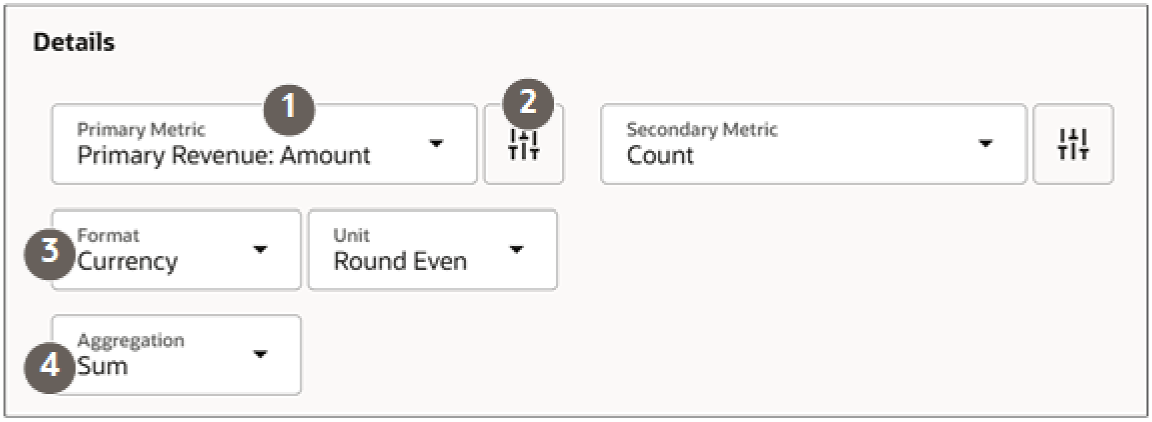
Create the Visualization

1. Make sure that the data you're using is visible to the user performing this configuration. You can temporarily remove filters.
2. Navigate to **Configuration** > **Application Composer**.
3. Click **Visualization Configuration** in the right-hand pane.
4. In the Visualization Configurations page, click **Add**.
5. From the Create Configuration page, **Source Type** field, select the **Adaptive Search** or **OTBI Analysis**, depending on the source of your data.
6. If you selected **Adaptive Search** as the source type, then:
   a. Select the object.
   b. In the **Saved Search** field, then select **Default** or a specific saved search.

      Selecting **Default** uses the default saved search for that object as your starting point and permits you to change the filters in the Filters region.



7. If you selected **OTBI Analysis**, then search for the analysis by name. By default, the application searches the Custom folder, but you can change the folder using the **Folders** button.
8. In the **Visualization Type** field, select **Metric Card**.
9. In the **Name** field, enter the title of the tab you're creating.
10. In the Details section, **Primary Metric** field, select the type of information you want to summarize under the title. For example, to display opportunity revenue, select Primary Revenue: Amount.
11. Click the **Preferences** button (broken vertical lines icon).
12. In the Format field, select how you want the data to be represented. For revenue, select Currency to display the currency symbol.
13. In the **Aggregation** field, select the calculation type. For revenue, select Sum, for example.



14. Repeat the same entries for any **Secondary Metric**. The secondary metric is the number which appears in parentheses next to the primary metric. For opportunity revenue, you may want to display the number of opportunities by selecting **Count**, for example.

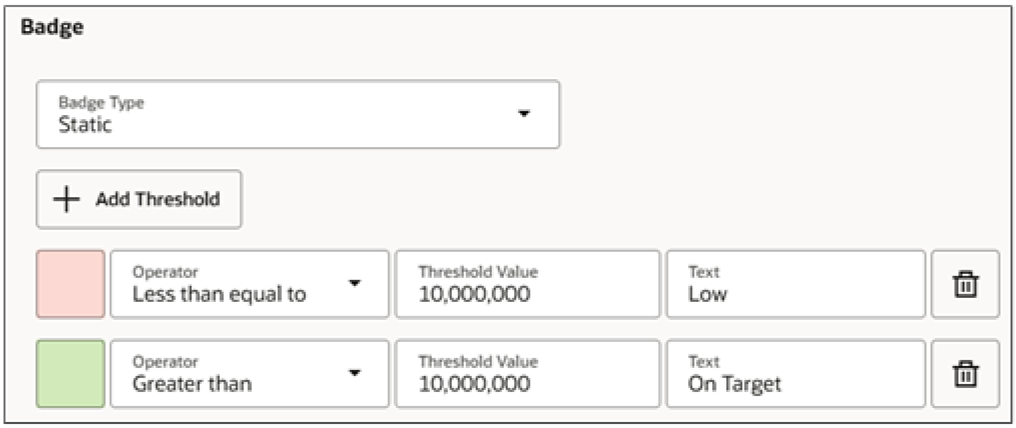
15. In the **Badge** section, specify the badges and the ranges of values that display them:

    For example, you can configure these badges for tasks:

    - If fewer than 5 overdue tasks exist, then don't show a badge at all.
    - If 5-10 overdue tasks exist, then show a warning badge.
    - If more than 10 overdue tasks exist, then show a critical badge.

    The order of the conditions is important. Each condition is evaluated in the order in which you define them, from top to bottom. Whichever condition is satisfied first is the condition that's applied.

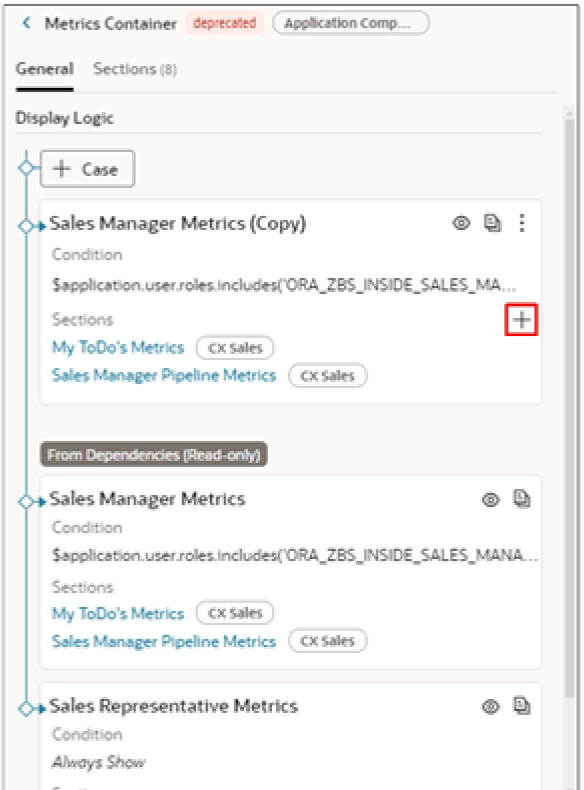
    a. In the **Badge Type** field, select **Static**.
    b. Add the thresholds. For each threshold, specify the badge color, operator, threshold value and short text.



16. Set the **Status** field to **Active**.
17. Click **Create** to save the configuration.
18. Note the number of the visualization configuration on the list page.

Create the Metric Card with the Visualization

1. Click **Add Section** in a metrics container copy. (The button is the plus sign highlighted in the screenshot).



2. Click **New Section** at the top of the list.

> **Note:** Although VB Studio allows it, you can't duplicate the metric cards provided by Oracle. You must create your own by adding a new section.

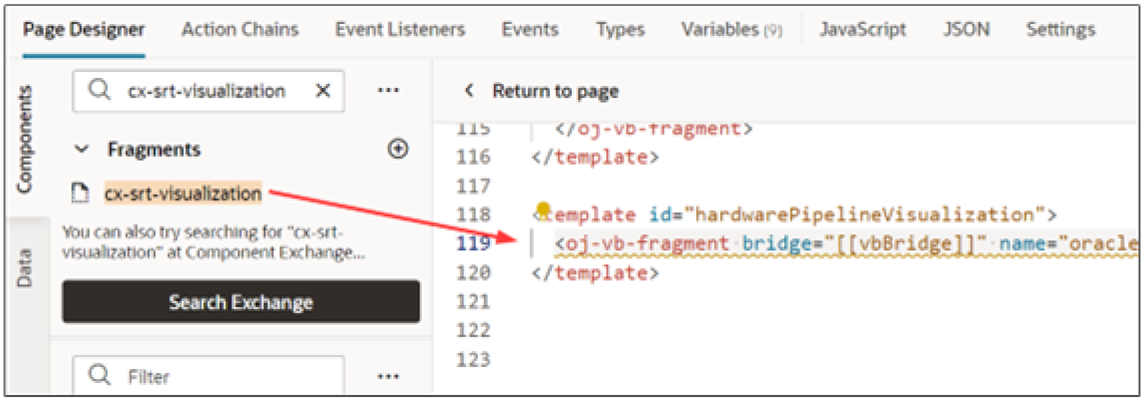
3. In the Create Section window, enter a title that's visible only during setup. For example, enter **Sales Manager Hardware Pipeline**. The title you enter here isn't the tab title that appears on the sales dashboard page. The runtime title comes from the visualization you created.
4. Drill down into the new section you created.
5. Click **Code**.

6. Your new **Hardware Pipeline** section displays empty placeholder `template` tags. For example:

```
<template id="salesManagerHardwarePipelineMetrics">
</template>
```

7. On the Components palette, in the Filter field, enter `cx-srt-visualization`.

8. Drag and drop the **cx-srt-visualization** fragment to the editor, between the template tags.



9. Add the following class information `class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height"` to the **<oj-vb-fragment line** before the **>** tag.

```
<oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-visualization"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height">
```

10. Below the bridge parameter and before the **</oj-vb-fragment>** tag, add the following fragment parameter. Enter the number of the visualization you created in the **value** parameter.

```
<oj-vb-fragment-param name="reportNumber" value="6001"></oj-vb-fragment-param>
```

11. Add the following parameters:

```
<oj-vb-fragment-param name="context"
 value='{"mode": "dashboard","source":"DV", "slot": "metric"}'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="filter" value="[[ $base.page.variables.dashboardFilters ]]"></oj-vb-
fragment-param>
 </oj-vb-fragment>
```

Your finished code will look like this:

```
<template id="hardwarePipeline">
 <oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-visualization"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height">
 <oj-vb-fragment-param name="reportNumber" value="6001"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="context"
 value='{"mode": "dashboard","source":"DV", "slot": "metric"}'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="filter" value="[[ $base.page.variables.dashboardFilters ]]"></oj-vb-
fragment-param>
 </oj-vb-fragment>
</template>
```

12. You can preview your changes by clicking the **Preview** button.
13. Click **Return to Page**.

You're now ready to configure the content container for the tab as described in the topic *Create the Contents Container and Link It to the Metric Card*.

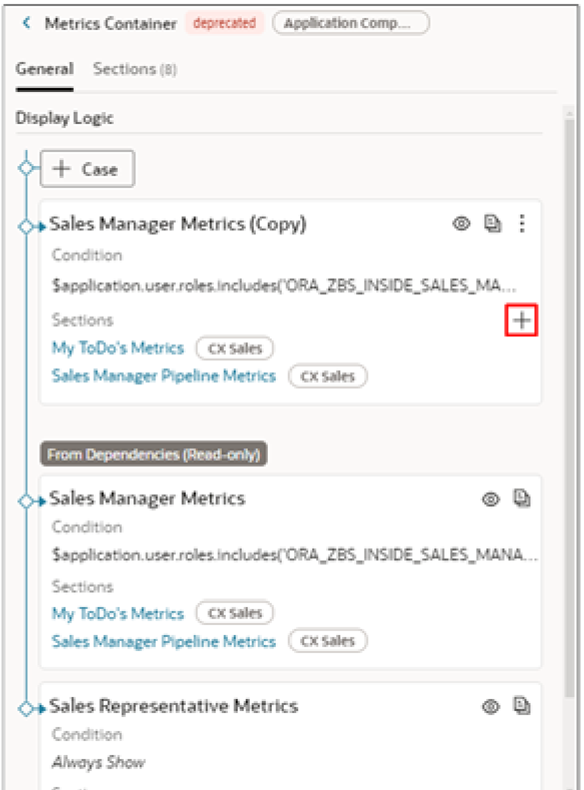
Create the Metric Card Directly in VB Studio

Instead of first creating the metric card as a visualization, you can create it directly in Oracle Visual Builder Studio.

You're now ready to configure the content container for the tab as described in the topic *Create the Contents Container and Link It to the Metric Card*.

Create the Metric Card

1. Click **Add Section** in a metrics container copy. (The button is the plus sign highlighted in the screenshot).



2. Click **New Section** at the top of the list.

> **Note:** Although VB Studio makes it possible to duplicate the metric cards provided by Oracle, you must instead add a new section, even if you're making a small change. You can't duplicate sections.

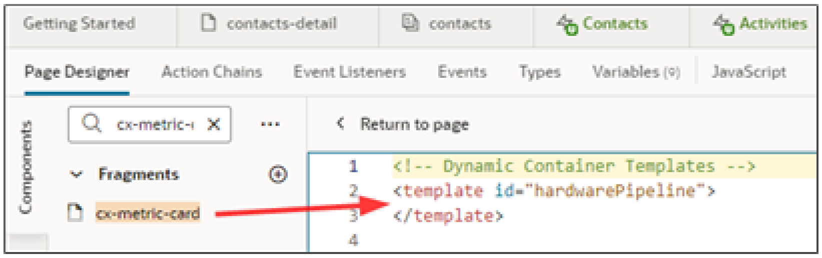
3. In the Create Section window, enter a title that's visible only during setup. You'll enter the actual tab title as a parameter in the code. For example, enter **Sales Manager Hardware Pipeline**.
4. Drill down into the new section you created.
5. Click **Code**.

6. Your new **Sales Manager Hardware Pipeline** section displays empty placeholder `template` tags.

```
<template id="salesManagerHardwarePipelineMetrics">
</template>
```

7. On the **Components** palette, in the **Filter** field, enter `cx-metric-card`.

8. Drag and drop the **cx-metric-card** fragment to the template editor, between the template tags.



You action inserts the following code:

```
<oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-metric-card"></oj-vb-fragment>
```

9. Add the following class information to the fragment: `class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height`.

   The resulting code looks like this:

```
<oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-metric-card" class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height></oj-vb-fragment>
```

10. Insert the following code for the metric card before the `</oj-vb-fragment>` tag

    a. Add the required **resource** parameter, which identifies the business object. The resource parameter is different for each object.

       Here's the resource parameter for opportunities, the object for pipeline information:

```
<oj-vb-fragment-param name="resource" value='[[ {"name": "opportunities", "primaryKey": "OptyId",
 "puid": "OptyNumber", "endpoint": "cx" } ]]'></oj-vb-fragment-param>
```

       For other supported business objects, see the list of resource parameters in *Values to Use in Code for Standard Components in Tables, Metric Cards, and List Views*.

    b. Add the required **query** parameter, which specifies the saved search that's the source of your data.

       In this example, the source is a saved search with the ID `caf84b9d-b975-4b9d-b78e-41718b78f36e`. You can obtain the ID of saved searches by inspecting the code in the Chrome browser as described the topic *How can I find the UUIDs for a saved search and the IDs for its filters in Oracle Sales in the Redwood User Experience?*.

```
<oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key":
"queryUuid", "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e"}]},
] ]]'></oj-vb-fragment-param>
```

    c. Optionally, you can filter the data from your source. For example, the sample parameters here filter the data to include only open opportunities owned by a manager's subordinates.

       > **Tip:** The saved search may already include filters, so you might want to add filters there instead of having another place to update in the future.

```
<oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key":
"queryUuid", "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e"}]},
{"type": "qbe", "params": [{"key": "StatusCode", "operator": "in", "value": ["OPEN"] },
{"key": "RecordSet", "value": "ORA_MYSUBORDSSALESTEAMOPTIES" }]},
```

```
] ]]'></oj-vb-fragment-param>
```

d. Optionally, add badges that draw attention to critical information, as described in the section *Add Badges to the Metric Card*.

e. Add the **title** parameter with the title that displays on the tab.

```
<oj-vb-fragment-param name="titleItem" value="[['Hardware Pipeline']]"></oj-vb-fragment-param>
```

f. To display the sum on the tab, add the **aggregate** parameter to calculate it. In this example, the parameter calculates the sum of the opportunity revenue in the saved search:

```
<oj-vb-fragment-param name="aggregate" value="[[ {'field':
 'PrimaryRevenue.RevnAmount','functionType': 'sum' } ]]"></oj-vb-fragment-param>
```

Here's sample code for the opportunity saved search without the optional filters and badges:

```
<template id="salesManagerHardwarePipelineMetrics">
 <oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-metric-card" class="oj-flex
 oracle-cx-fragmentsUI-cx-fragment-full-height">
 <oj-vb-fragment-param name="resource" value='[[ {"name": "opportunities", "primaryKey": "OptyId",
 "puid": "OptyNumber", "endpoint": "cx" } ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e"}]},
 ] ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="aggregate" value="[[ {'field': 'PrimaryRevenue.RevnAmount','functionType':
 'sum' } ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="titleItem" value="[['Hardware Pipeline']]"></oj-vb-fragment-param>
 </oj-vb-fragment>
</template>
```

11. Click **Return to Page**.

Add Badges to the Metric Card

First, determine the display conditions. Each condition is a range of aggregate sum numbers that display on the metric card. For each range, you decide which type of badge to display. For example, you can configure this type of setup:

- If fewer than 5 overdue tasks exist, then don't show a badge at all.
- If 5-10 overdue tasks exist, then show a warning badge.
- If more than 10 overdue tasks exist, then show a critical badge.

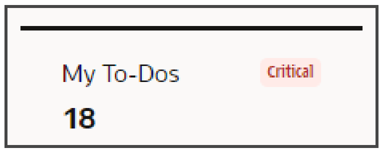
When you define the display conditions for a metric card, the order of those conditions is important. Each condition is evaluated in the order in which you define them, from top to bottom. Whichever condition is satisfied first is the condition that's applied.

You can also configure the metric card so that the card always displays the same badge regardless of the aggregate sum on the card.

The following table lists the badge properties that you can use to define a set of badges and display conditions:

Badge Properties

Badge Property	Description
`status`	• danger

Badge Property	Description
	• info • neutral • success • warning Each type of badge (**`status`**) displays a specific color, which you can't change. Here's an example of a "danger" badge with the text "Critical": 
`min`	Lower limit of the display condition for a specific badge.
`max`	Upper limit of the display condition for a specific badge.
`text`	Text that displays on the badge. Each type of badge (**`status`**) displays with default text, but you can use this property to change it.

Here's a sample of the code that you can use to define a set of badges and display conditions:

```
[{status : "success", text : "All Good", min : "0", max : "10"}, {status : "warning", text : "Attention",
 min : "11", max : "50"},{status : "danger", text : "Critical", min : "51"}]
```

- Specify the type of badge to display (**`status`**).
- Define the conditions (**`min`** and **`max`**) that display each badge and an optional description.
- Indicate the badge text (**`text`**), for example, **Warning** or **Problem**.

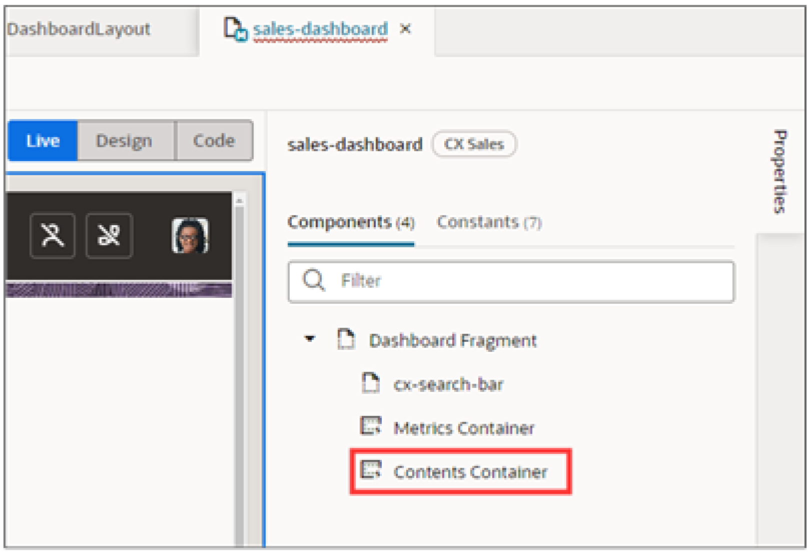
Tip:

- You can specify only a lower limit without an upper limit (to indicate an aggregate sum that's greater than 10, for example).
- You can add **`metaText`** without any range conditions. So, if none of the ranges apply, then show a description on the metric card.

Create the Contents Container and Link It to the Metric Card

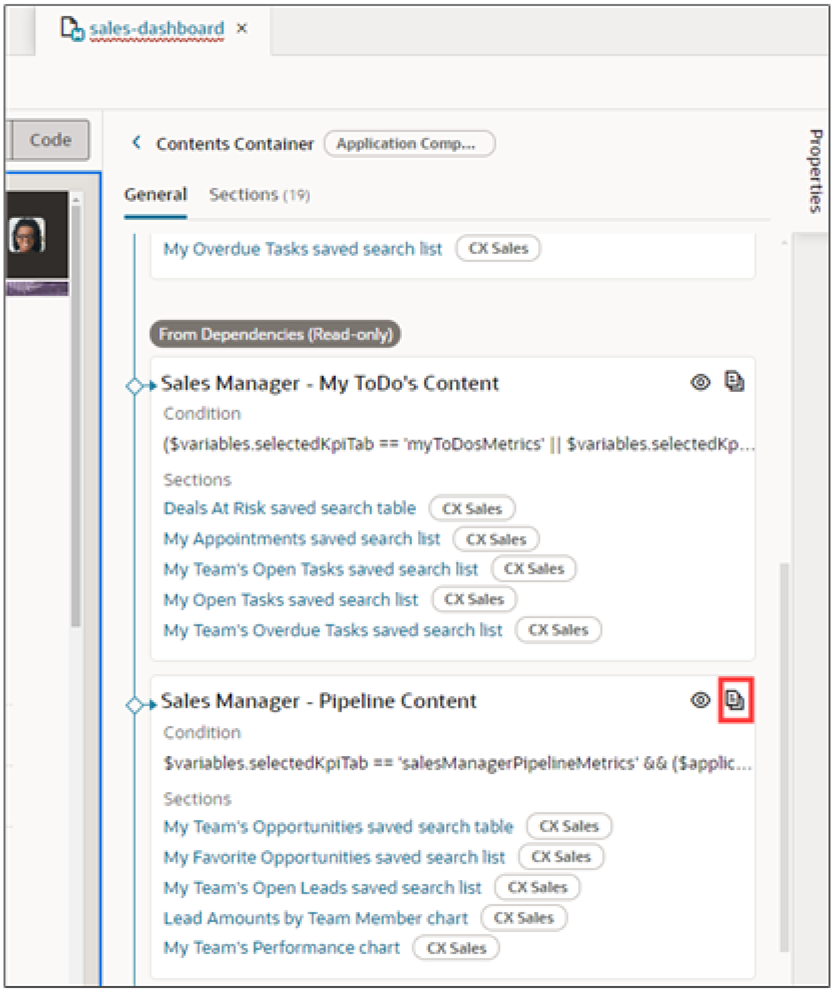
After you've created a custom metric card for your dashboard tab, you're ready to create the container that will hold the content you want to display. Here's how.

1. Open the Sales Dashboard and edit the page in Oracle Visual Builder Studio.
2. On the Components pane, click **Contents Container**.



3. Oracle provides a contents container for every metrics card, so find the one that includes content of interest and click, **Duplicate**.

   In this example, we're creating a tab that displays information on the team hardware opportunity pipeline, so we'll duplicate the **Sales Manager - Pipeline Content**.



4. Edit the **Condition** field and replace the ID with the ID for the metrics card you created and click **Confirm** (the check mark icon). For example:

```
$base.variables.selectedKpiTab == 'salesManagerHardwarePipelineMetrics'
```

   This link ensures the contents of your tab displays each time a user clicks on the metric card.

You're now ready to configure the contents to the tab as described in the topic *Add the Sections to the Tab's Contents Container*.

Add the Sections to the Tab's Contents Container

You can add up to 5 sections in a tab. Each section can display a table, a list, a visualization, or embedded Oracle Fusion Intelligence content.

> **Tip:** Before you start, read the topic *How Information Displays in a Sales Dashboard Tab as You Add Sections* to familiarize yourself with the content styles that determine how tabs are laid out as you add sections. If you add just one section, for example, that section always takes up the whole tab. As you add more sections using the default layout style, the first panel always remains bigger that the rest, but shrinks to make room for the others.

1. If the copy of the content container already includes 5 sections, then remove one. To delete a section, select a row and click **Remove Item** (the trash can icon).
2. In the custom content container, click **Add Section**.
3. To add any of the content predefined by Oracle as is, scroll down and select it from the list. The available content includes tables, lists, and 2 visualizations. See the *Predefined Content That You Can Add to Your Dashboard Tabs* topic for details.
4. Here's how to add custom content:

   a. Click **New Section**.
   b. In the Create Section window, enter a title for the section, for example: **Hardware Opportunities Pipeline**. The title is visible only during setup and as the ID in the code.
   c. Drill down into the copy.
   d. Click **Code**.

   The section you added displays with placeholder `template` tags.

   e. Enter the appropriate code between the template tags. The code differs by content type.

      - Follow these steps for all visualizations, including tables that you created as visualizations: *Add a Custom Visualization to Your Dashboard Tab*
      - Follow these steps to create a table directly in VB Studio: *Display a Saved Search as a Table If You Didn't Create It As a Visualization*
      - *Display a Saved Search as a List*
      - *Add My List*
      - *Embed Content Created in Oracle Fusion Data Intelligence*

Display a Saved Search as a Table If You Didn't Create It As a Visualization

Use these steps to display a table from a saved search if you didn't create the table as a visualization. If your table is a visualization, then follow the steps for adding visualizations instead:*Add a Custom Visualization to Your Dashboard Tab*.

1. On the **Components** palette, in the **Filter** field, enter `cx-subview`.
2. Drag and drop the **cx-subview** fragment to the template editor, between the template tags.

3. Add the following class information: `class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height"` to the **<oj-vb-fragment line** before the **>** tag.

```
<oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-subview"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height">
```

4. Add the required **resource** parameter, which specifies the business object you're displaying. Each object requires different values. Here are the values for opportunities:

```
<oj-vb-fragment-param name="resource"
 value='[[ {"name": "opportunities", "primaryKey": "OptyId", "puid": "OptyNumber", "endpoint":
 "cx" } ]]'>
 </oj-vb-fragment-param>
```

   The values for other business objects are listed in the topic *Values to Use in Code for Standard Components in Tables, Metric Cards, and List Views*.

5. If you're displaying a table with opportunity revenue, you must insert this additional parameter because opportunity revenue is a child object of opportunity:

```
<oj-vb-fragment-param name="child" value='[[ {"name": "ChildRevenue", "primaryKey": "RevnId"} ]]'></oj-
vb-fragment-param>
```

6. Add the required **query** parameter, which specifies the saved search that is the source of the data in your table:

```
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e" }]}
 ] ]]'></oj-vb-fragment-param>
```

   In this example, the source is a saved search with the ID `caf84b9d-b975-4b9d-b78e-41718b78f36e`. You can obtain the ID of saved searches by inspecting the code in the Chrome browser as described the topic *How can I find the UUIDs for a saved search and the IDs for its filters in Oracle Sales in the Redwood User Experience?*.

7. Optionally, you can filter the data from your source. For example, the sample parameters here filter the data to include only open opportunities owned by a manager's subordinates.

```
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e"}]},
 {"type": "qbe", "params": [{"key": "StatusCode", "operator": "in", "value": ["OPEN"] },
 {"key": "RecordSet", "value": "ORA_MYSUBORDSSALESTEAMOPTIES" }]},
```

```
] ]]'></oj-vb-fragment-param>
```

8. Add the following **style** parameter. It specifies the size and position of the sections in the tab as you add more sections. For details see the topic *How Information Displays in a Sales Dashboard Tab as You Add Sections*

```
<oj-vb-fragment-param name="style" value="[[ 'dashboard' ]]"></oj-vb-fragment-param>
```

9. Add the **subviewLayoutId** parameter to use the dashboard layout (**dashboardLayout**). The dashboard layout specifies which columns are displayed in the table, in which order, and in which relative widths. The default layout already includes the important columns, but you can create your own custom layouts as described in the topic: *Specify the Columns in a Sales Dashboard Table and Their Widths*

```
<oj-vb-fragment-param name="subviewLayoutId" value="[[ 'dashboardLayout' ]]"></oj-vb-fragment-param>
```

10. Add the **title** parameter, to specify the title that displays on the tab.

```
<oj-vb-fragment-param name="title" value="[['Hardware Opportunities]]"></oj-vb-fragment-param>
```

Sample Code for Opportunity Pipeline Table from a Saved Search

Here's the complete code for an opportunity table (leaving out the optional query parameters):

```
<template id="hardwareOpportunitiesInPipeline">
 <oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-subview"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height">
 <oj-vb-fragment-param name="resource"
 value='[[ {"name": "opportunities", "primaryKey": "OptyId", "puid": "OptyNumber", "endpoint": "cx" } ]]'>
 </oj-vb-fragment-param>
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e" }]}
 ] ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="style" value="[[ 'dashboard' ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="title" value="[['Hardware Opportunities']]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="subviewLayoutId" value="[[ 'dashboardLayout' ]]"></oj-vb-fragment-param>
 </oj-vb-fragment>
</template>
```

Sample Code for a Table with Opportunity Revenue

To add a table of opportunity revenue, you must add an additional revenue parameter because a revenue line is a child object of opportunity:

```
<oj-vb-fragment-param name="child" value='[[ {"name": "ChildRevenue", "primaryKey": "RevnId"} ]]'></oj-vb-
fragment-param>
```

Here is sample code with the child fragment added:

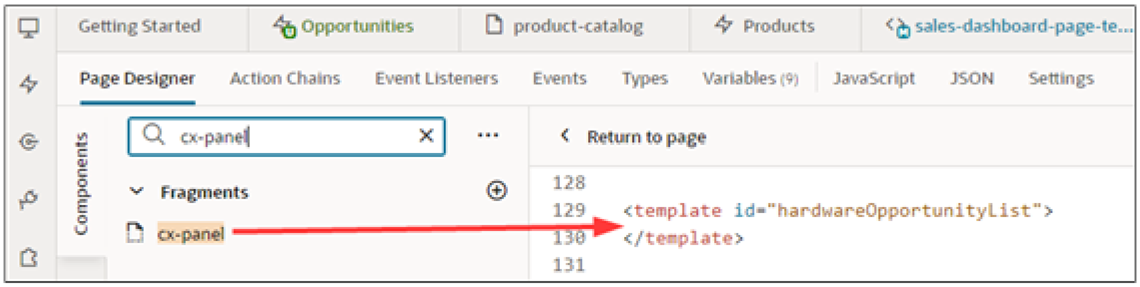
```
<template id="revenueTable">
<oj-vb-fragment name="oracle_cx_fragmentsUI:cx-subview" class="oj-flex oracle-cx-fragmentsUI-cx-fragment-
full-height" bridge="[[ vbBridge ]]">
 <oj-vb-fragment-param name="resource" value='[[ {"name": "opportunities", "primaryKey": "OptyId", "puid":
 "OptyNumber", "endpoint": "cx" } ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="child" value='[[ {"name": "ChildRevenue", "primaryKey": "RevnId"} ]]'></oj-vb-
fragment-param>
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "73b21b33-db08-4327-bf30-88c3c9e0f70d" }]}] ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="style" value="[[ 'dashboard' ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="title" value="[['Revenue']]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="subviewLayoutId" value="[[ 'dashboardLayout' ]]"></oj-vb-fragment-param>
 </oj-vb-fragment>
```

```
</template>
```

Display a Saved Search as a List

You can display saved searches as lists in the Sales Dashboard. Here's how.

1. On the Components palette, in the **Filter** field, enter `cx-panel`.
2. Drag and drop the **cx-panel** fragment to the template editor, between the template tags.



3. Add the following class information `class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height"` to the **<oj-vb-fragment** line before the **>** tag. Here's the resulting line:

```
<oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-panel"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height">
```

4. Add the required **resource** parameter, which specifies the object you're displaying. Here are the values for opportunities:

```
<oj-vb-fragment-param name="resource"
 value='[[ {"name": "opportunities", "primaryKey": "OptyId", "puid": "OptyNumber", "endpoint":
 "cx" } ]]'>
 </oj-vb-fragment-param>
```

   Each object requires different values, which are listed in the topic *Values to Use in Code for Standard Components in Tables, Metric Cards, and List Views*.

5. If you're displaying a list with opportunity revenue, you must insert this additional parameter because opportunity revenue is a child object of opportunity:

```
<oj-vb-fragment-param name="child" value='[[ {"name": "ChildRevenue", "primaryKey": "RevnId"} ]]'></oj-
vb-fragment-param>
```

6. Add the required **query** parameter, which specifies the saved search that's the source of the data in your list:

```
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e" }]},
 $base.page.variables.dashboardFilters] ]]'></oj-vb-fragment-param>
```

   In this example, the source is a saved search with the ID `caf84b9d-b975-4b9d-b78e-41718b78f36e`. You can obtain the ID of saved searches by inspecting the code in the Chrome browser as described the topic *How can I find the UUIDs for a saved search and the IDs for its filters in Oracle Sales in the Redwood User Experience?*.

7. Optionally, you can sort the list. For example, enter the following to sort the records in ascending order by the last update.

```
<oj-vb-fragment-param name="sortCriteria" value='[[ [{"attribute": "LastUpdateDate","direction":
 "asc" }] ]]'></oj-vb-fragment-param>
```

8. Add the following **style** parameter. It applies the style for the dashboard layout. For details see the topic *How Information Displays in a Sales Dashboard Tab as You Add Sections*

```
 <oj-vb-fragment-param name="style" value="[[ 'dashboard' ]]"></oj-vb-fragment-param>
```

9. Add the **subviewLayoutId** parameter to use the dashboardCardLayout. The dashboard card layout is where you specify which fields you want to display in your list. Each object already includes the important fields.

```
<oj-vb-fragment-param name="panelCardLayoutId" value="[[ 'dashboardCardLayout' ]]"></oj-vb-fragment-
param>
```

10. Add the **title** parameter, to specify the title that displays on the list.

```
<oj-vb-fragment-param name="title" value="[['Hardware Opportunities]]"></oj-vb-fragment-param>
```

Sample Code for Displaying a List of Opportunities

```
<template id="hardwareOpportunityList">
 <oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-panel"class="oj-flex oracle-cx-
fragmentsUI-cx-fragment-full-height">
 <oj-vb-fragment-param name="resource"
 value='[[ {"name": "opportunities", "primaryKey": "OptyId", "puid": "OptyNumber", "endpoint": "cx" } ]]'>
 </oj-vb-fragment-param>
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "caf84b9d-b975-4b9d-b78e-41718b78f36e" }]},
 $base.page.variables.dashboardFilters] ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="sortCriteria" value='[[ [{"attribute": "LastUpdateDate","direction":
 "asc" }] ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="style" value="[[ 'dashboard' ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="panelCardLayoutId" value="[[ 'dashboardCardLayout' ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="title" value="[['Hardware Opportunities']]"></oj-vb-fragment-param>
 </oj-vb-fragment>
 </template>
```

Sample Code for a List Of Opportunities with a Revenue Component

For an opportunity list with a revenue component, add the child parameter:

```
 <oj-vb-fragment-param name="child" value='[[ {"name": "ChildRevenue", "primaryKey": "RevnId"} ]]'></oj-vb-
fragment-param>
```

Here's the sample code:

```
<template id="revenueList">
<oj-vb-fragment name="oracle_cx_fragmentsUI:cx-panel" class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-
height" bridge="[[ vbBridge ]]">
 <oj-vb-fragment-param name="resource" value='[[ {"name": "opportunities", "primaryKey": "OptyId", "puid":
 "OptyNumber", "endpoint": "cx" } ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="child" value='[[ {"name": "ChildRevenue", "primaryKey": "RevnId"} ]]'></oj-vb-
fragment-param>
 <oj-vb-fragment-param name="query" value='[[ [{"type": "savedSearch", "params": [{"key": "queryUuid",
 "value": "73b21b33-db08-4327-bf30-88c3c9e0f70d" }]}] ]]'></oj-vb-fragment-param>
 <oj-vb-fragment-param name="extensionId" value="{{ 'oracle_cx_salesUI' }}"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="style" value="[[ 'dashboard' ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="title" value="[['Revenue']]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="panelCardLayoutId" value="[[ 'dashboardCardLayout' ]]"></oj-vb-fragment-param>
```

```
    </oj-vb-fragment>
  </template>
```

Add a Custom Visualization to Your Dashboard Tab

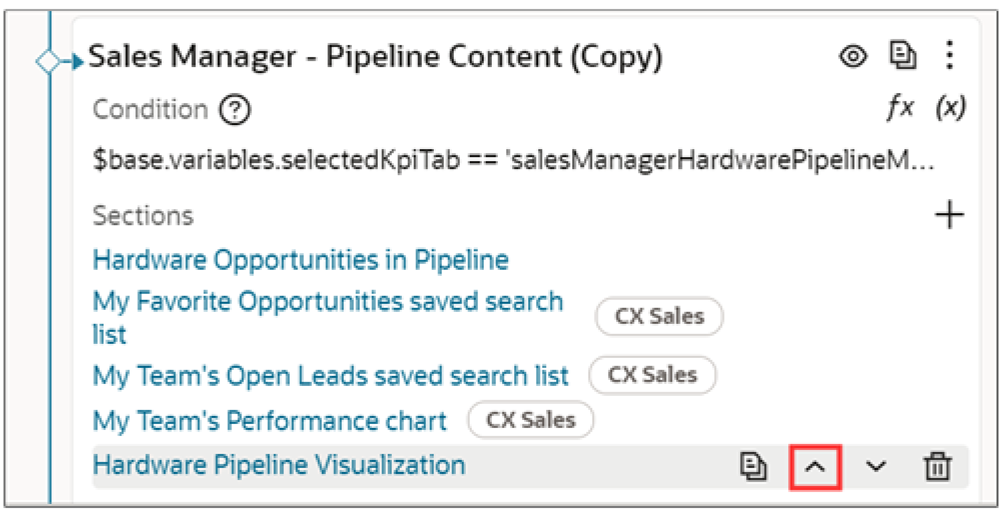
Here's how to add visualizations. These can be visualizations from Express Reports or those that you created using the Visualization Configurations.

> **Note:** You can only have up to 5 sections in each tab. If you're adding a visualization and the tab already includes 5 sections, then you must remove a section first.

1. On the Contents Container Properties pane, find the custom contents container where you want to add the visualization.
2. Click the **Add Section** (+) icon and **+ New Section**.
3. In the Create Section dialog, **Title** field, enter a title for the component and click **OK**.

   > **Note:** The title is the name of the component and doesn't appear on the dashboard at runtime. The runtime title comes from the `dashboardTitle` parameter that you add to the visualization code.

   The newly added section is added to the bottom of the layout. Use the **Move Up** arrow to move it to the desired location. For example, if you display a table showing a list of hardware opportunities in the pipeline, you may want to show a visualization of the opportunities by sales stage right next to it.

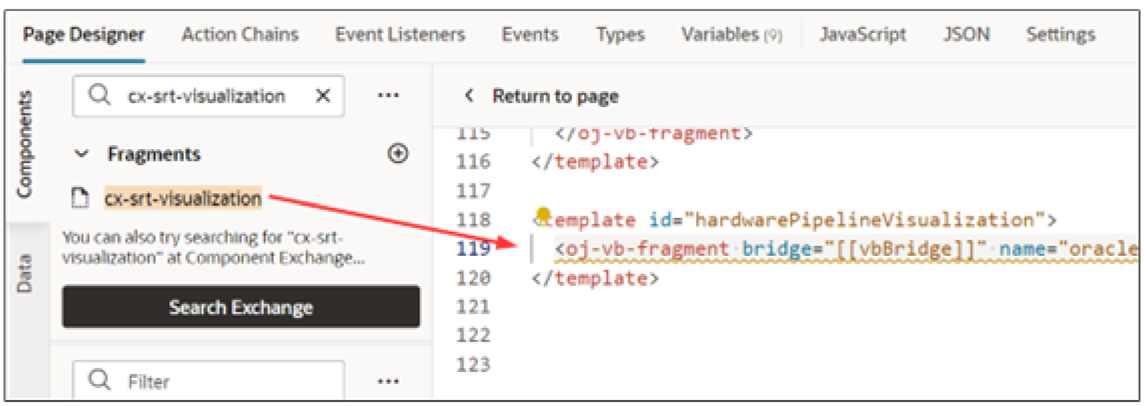
   

4. Click the section name link.

5. Click the **Code** button.

Your new section displays empty placeholder `template` tags.

6. On the Components palette, in the Filter field, enter `cx-srt-visualization`.
7. Drag and drop the **cx-srt-visualization** fragment to the editor, between the template tags.



8. Add the following class information `class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height"` to the **<oj-vb-fragment line** before the **>** tag.

```
<oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-visualization"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height">
```

9. Add the following fragment parameters to the code below the bridge parameter and before the **</oj-vb-fragment>** tag.

```
<oj-vb-fragment-param name="reportNumber" value="XXXX"></oj-vb-fragment-param>
<oj-vb-fragment-param name="context" value='{"mode":
"dashboard","source":"DV","dashboardTitle":"XXXXX"}'></oj-vb-fragment-param>
```

```
 <oj-vb-fragment-param name="filter" value="[[ $base.page.variables.dashboardFilters ]]"></oj-vb-
fragment-param>
```

In your fragment code, replace the values for the `reportNumber`, `source`, and `dashboardTitle` parameters with the values for your custom visualization.

This table describes the key parameters:

Custom Visualization Parameters

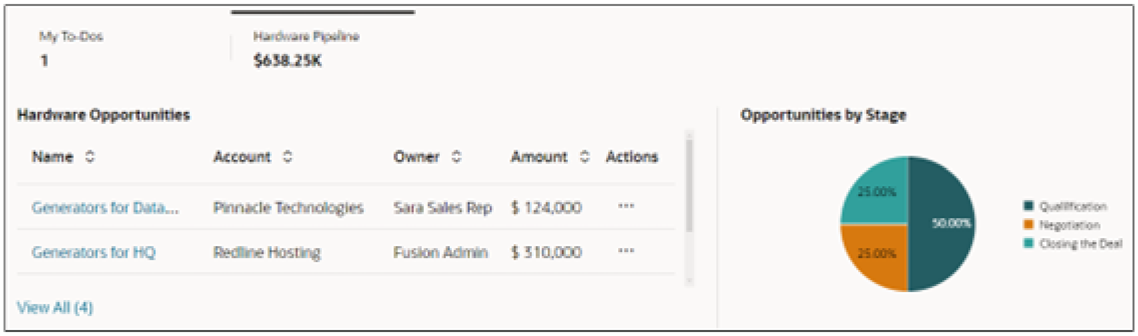
Parameter Name	Description
reportNumber	Enter the reference number of the visualization that you created using the Visualization Configuration tool. You can also display the visualizations created in Express Reports, by entering its report number. The sales dashboard only displays the visualization part of the Express Report. The table isn't displayed.
mode	This value should always be `dashboard`.
source	Enter `DV` for visualizations you created using the Visualization Configuration tool. To display the visualizations created in Express Reports, enter `SRT`.
dashboardTitle	Specify the title of the visualization that displays above the chart at runtime.

Here's a finished code sample for a visualization created in the Visualization Configuration tool:

```
<template id="hardwarePipelineVisualization">
 <oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-visualization"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height">
 <oj-vb-fragment-param name="reportNumber" value="4001"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="context"
 value='{"mode": "dashboard","source":"DV", "dashboardTitle":"Opportunities by Stage"}'></oj-vb-
fragment-param>
 <oj-vb-fragment-param name="filter" value="[[ $base.page.variables.dashboardFilters ]]"></oj-vb-
fragment-param>
 </oj-vb-fragment>
</template>
```

10. Click **Preview** to view the result.

    Here's a sample visualization of hardware opportunities by sales stage next to the table of hardware opportunities.



Add My List

Here's how to add the **My List** component to the Sales Dashboard. My List includes data from all objects. For example, My List can include data from both opportunities and leads.

Here's the sample fragment code:

```
<template id="myList">
<oj-vb-fragment name="oracle_cx_fragmentsUI:cx-heterogeneous-list"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height" bridge="[[ vbBridge ]]">
 <oj-vb-fragment-param name="query" value="[[ [{type: 'qbe', params: [{key: '_userRelevantItems', value:
 true}]}] ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="style" value="[[ 'dashboard' ]]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="title" value="[['My List']]"></oj-vb-fragment-param>
 <oj-vb-fragment-param name="entities" value="[[ ['accounts','contacts','leads','opportunities'] ]]"></oj-
vb-fragment-param>
</oj-vb-fragment>
</template>
```

In your fragment code, be sure to update the values in the "entities" and "title" parameters, as necessary.

Embed Content Created in Oracle Fusion Data Intelligence

Here's how you can embed content created in Oracle Fusion Data Intelligence.

Here's an example of the type of content you can embed:

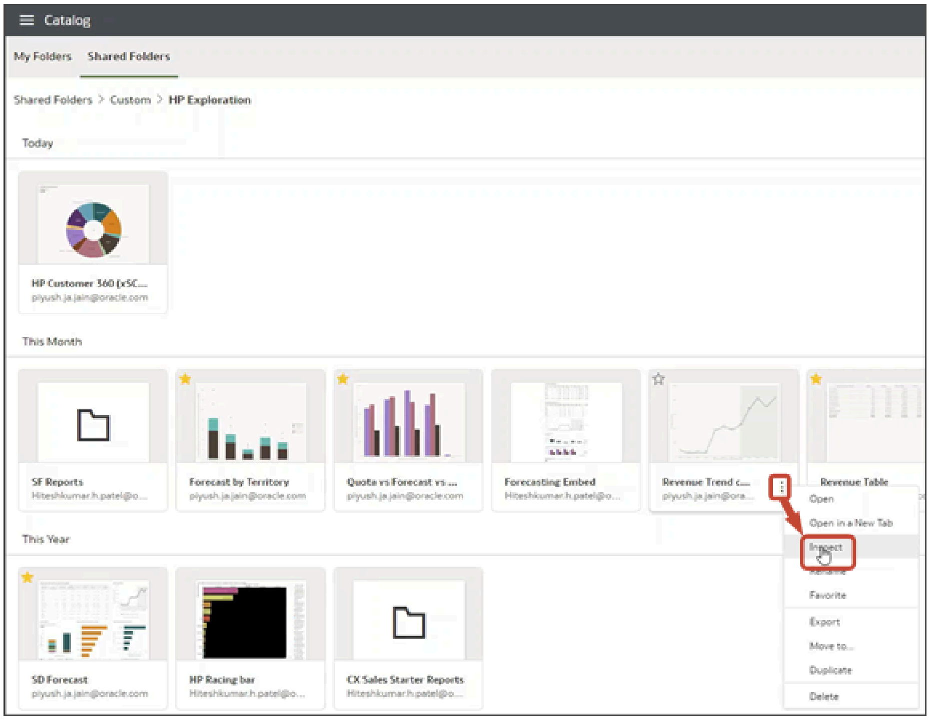
For more information about the Oracle Fusion Data Intelligence embedding framework, see the *Visualizing Data and Building Reports in Oracle Analytics Cloud* guide.

Prerequisites

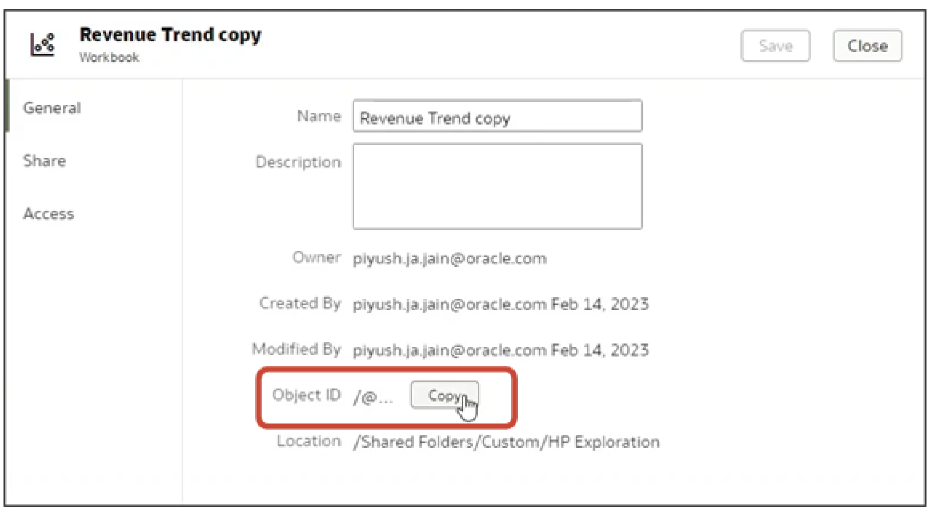
Before adding an Oracle Fusion Data Intelligence (OFDI) component to the Sales Dashboard, you must retrieve the Object ID from your existing OFDI content:

1. Navigate to the OFDI host where the analytics workbook is saved.

2. Click the **Actions** > **Inspect**.



3. Next to the Object ID, click **Copy**.



4. Paste the Object ID, along with the Fusion Data Intelligence host URL, into a separate file for later use. For example:

   a. Oracle Analytics Host:

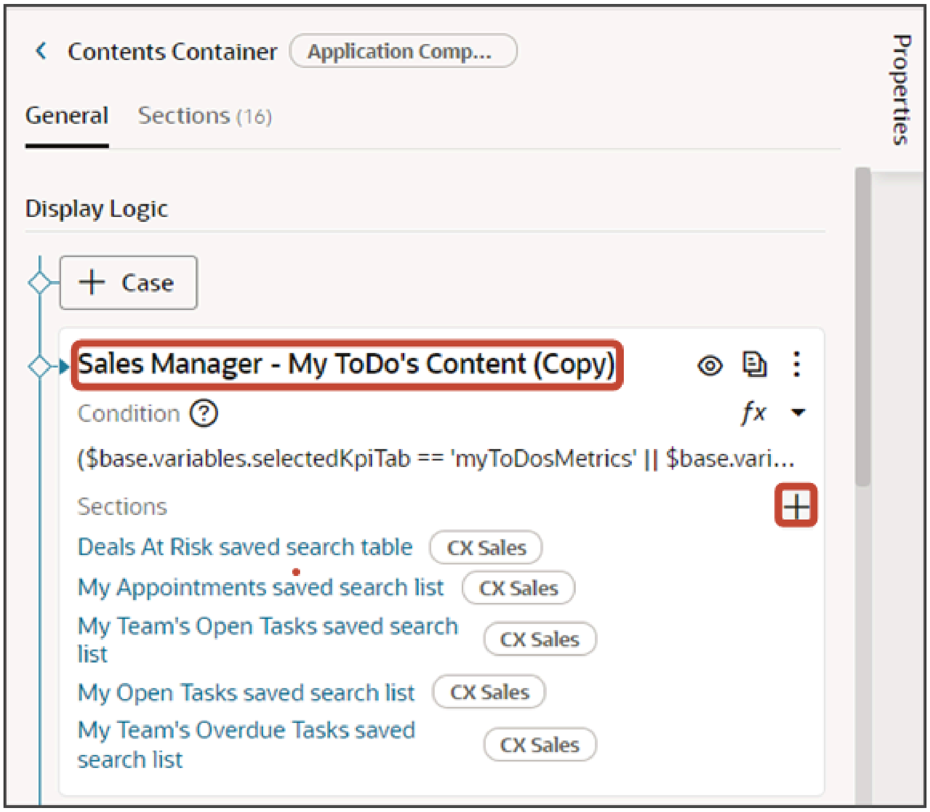
   ```
   https://oac01-gse00010001-px.analytics.ocp.oraclecloud.com/
   ```

   b. Object ID:

   ```
   /@Catalog/shared/Custom/HP Exploration/Revenue Trend copy
   ```

Add an Oracle Fusion Data Intelligence component to the Sales Dashboard

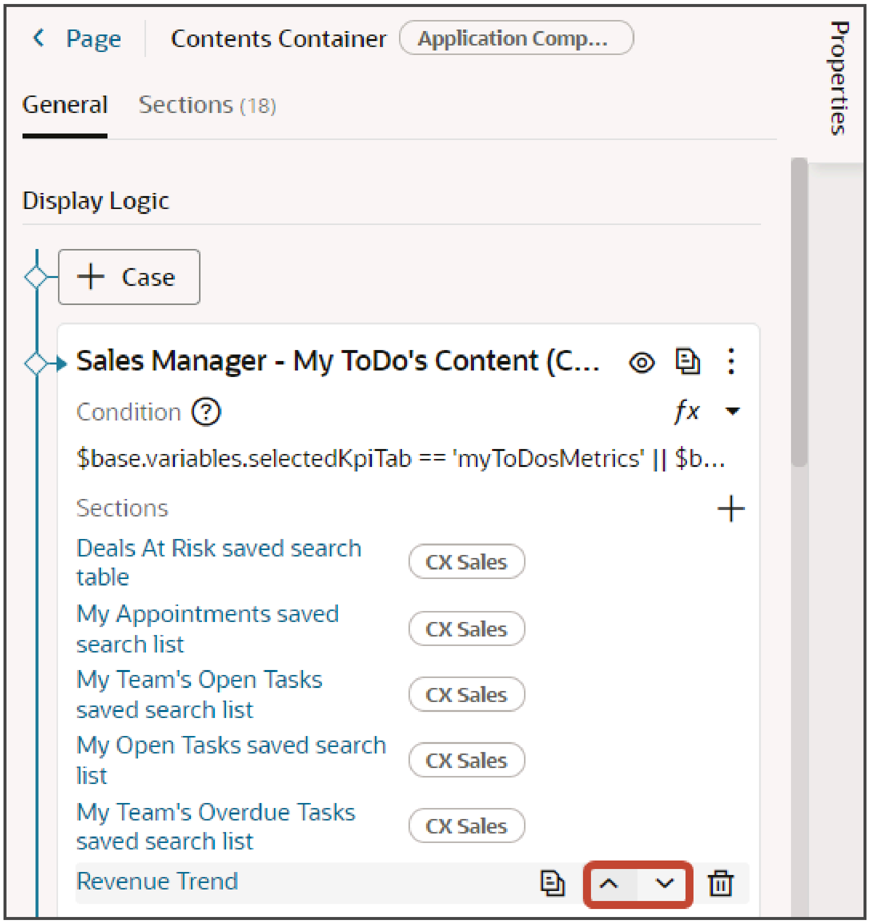
1. On the Properties pane for the **Sales Manager - My ToDo's Content (Copy)** layout, click the **+** Add Section icon.



2. In the Create Section dialog, in the Title field, enter `Revenue Trend`, and click **OK**.

> **Note:** The title is the name of the component, not how the component will appear at runtime on the dashboard. The runtime title comes from the `Panel Title` property that you specify for the Dashboard Panel fragment. See step 10.

The newly added section is added to the bottom of the layout. Use the Move Up arrow to move the new section to the desired location.

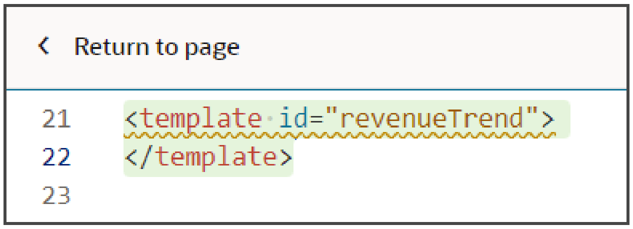


3. Click **Revenue Trend**.

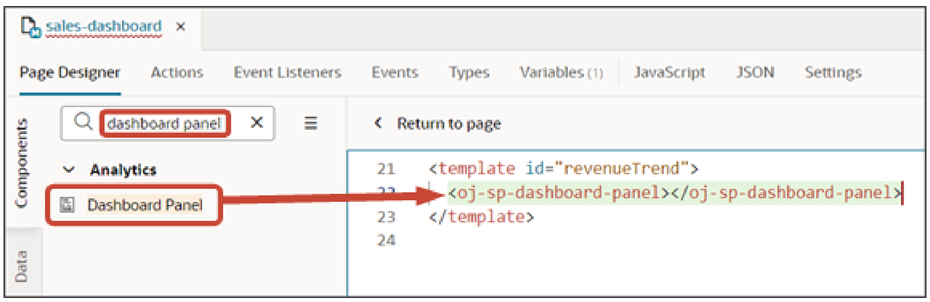
The template editor opens.

4. Click the **Code** button.

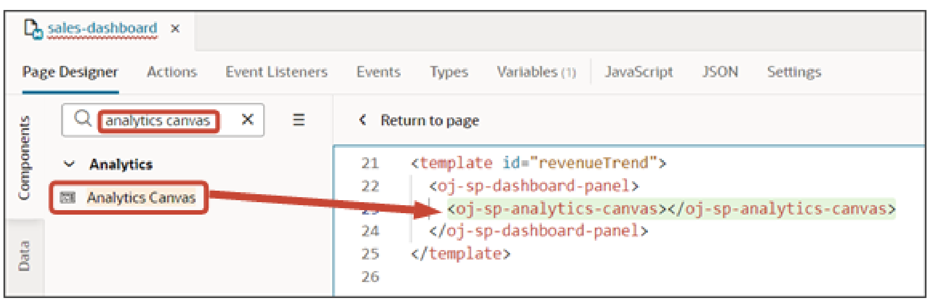
Your new **Revenue Trend** section displays with empty placeholder `template` tags.



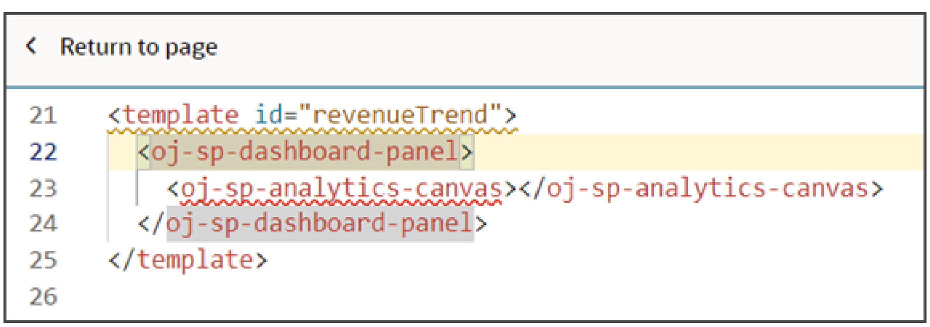
5. On the Components palette, in the **Filter** field, enter `dashboard panel`.
6. Drag and drop the **Dashboard Panel** fragment to the template editor, between the template tags.



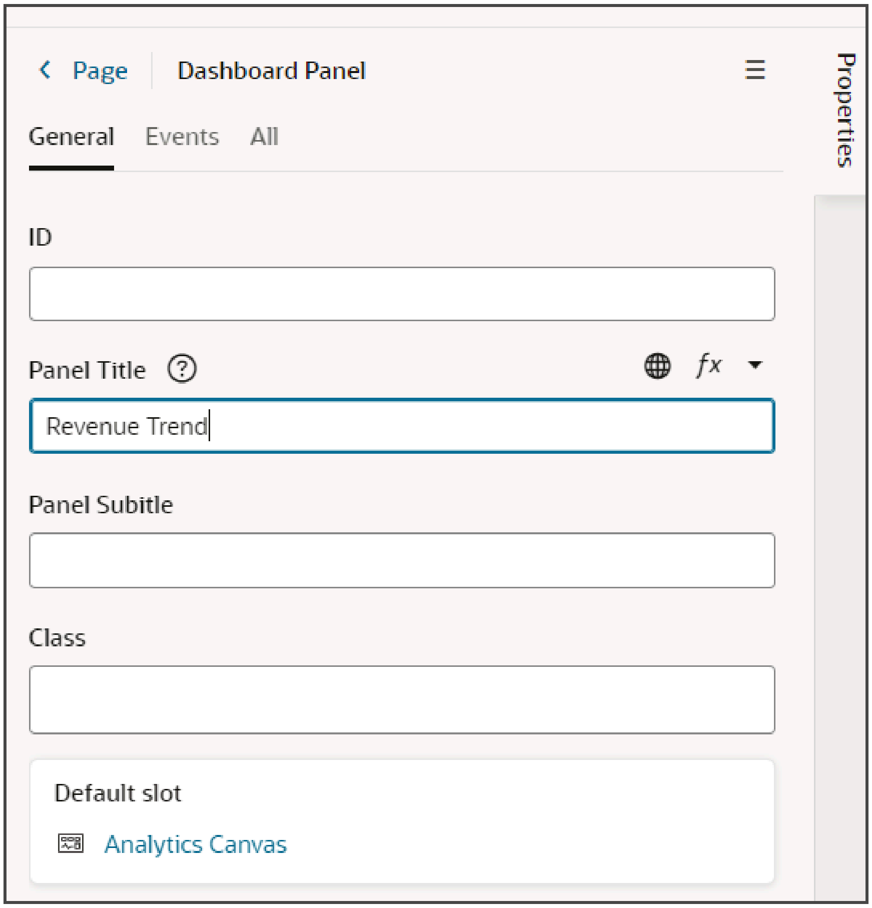
7. On the Components palette, in the **Filter** field, enter `analytics canvas`.
8. Drag and drop the **Analytics Canvas** fragment to the template editor, between the `oj-sp-dashboard-panel` template tags.



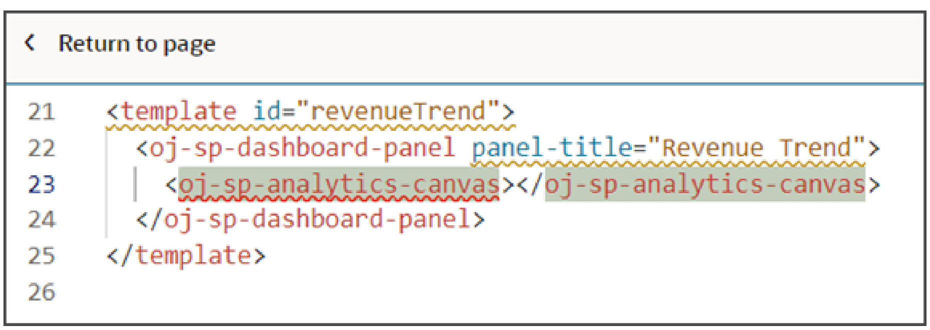
9. In the template editor, click the `oj-sp-dashboard-panel` template tags.



10. On the Properties pane, in the **Panel Title** field, enter the title that you want to display on the dashboard for the analytic, such as `Revenue Trend`.



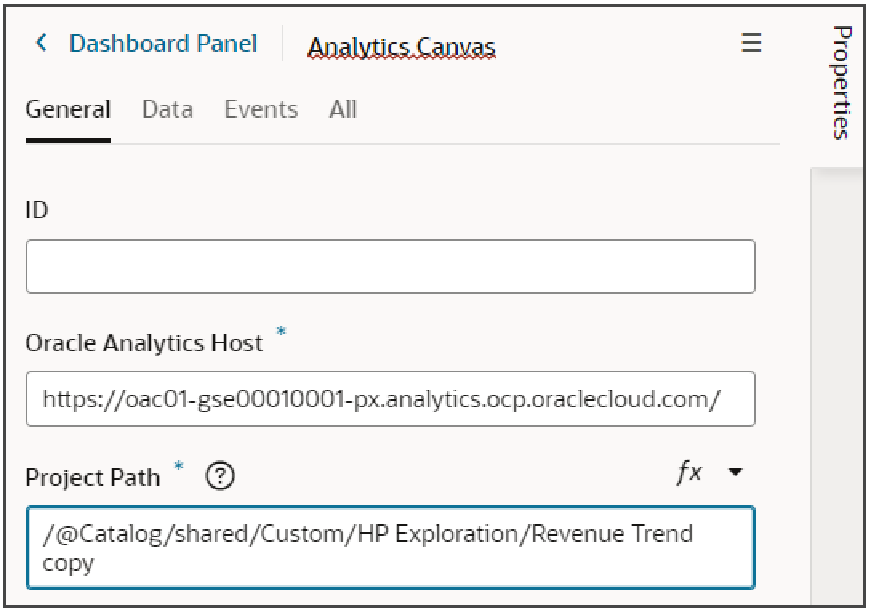
11. In the template editor, click the `oj-sp-analytics-canvas` template tags.



12. On the Properties pane, in the **Oracle Analytics Host** field, enter the host URL that you saved, such as `https://oac01-gse00010001-px.analytics.ocp.oraclecloud.com/`.

13. In the Project Path field, enter the repository path of the project to render, such as `/@Catalog/shared/Custom/HP Exploration/Revenue Trend copy`.

    Note: This is the Object ID of the Oracle Analytics Cloud workbook that you saved.



14. Test the modified dashboard layout by previewing your application extension.
    a. Click the **Preview** button to see your changes in your runtime test environment.

    b. The resulting preview link will be:

    `https://<servername>/fscmUI/redwood/cx-sales/dashboards/sales-dashboard`

    c. Change the preview link as follows:

    `https://<servername>/fscmUI/redwood/cx-sales/application/container/dashboards/sales-dashboard`

Preview Your Changes and Save Them to the Git Repository

As you configure your sales dashboard, you can preview your changes and then save them to the Git repository.

1. Test your sales dashboard edits, by previewing your application extension:
    a. Click the **Preview** button to see your changes in your runtime test environment.

b. The resulting preview link will be:

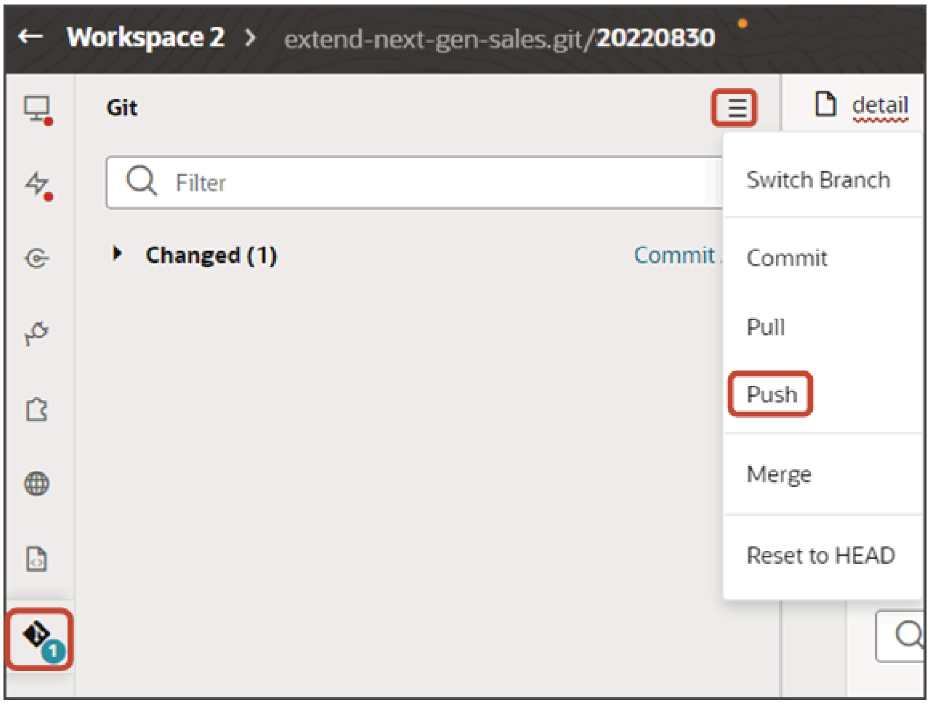
```
https://<servername>/fscmUI/redwood/cx-sales/dashboards/sales-dashboard
```

c. Change the preview link as follows:

```
https://<servername>/fscmUI/redwood/cx-sales/application/container/dashboards/sales-dashboard
```

2. If you've
3. Save your work by using the **Push Git** command.

   Navigate to the Git tab, review your changes, and do a Git push (which does both a commit and a push to the Git repository).



How Information Displays in a Sales Dashboard Tab as You Add Sections

A sales dashboard can include anywhere from 1 to 5 sections in each of the tab pages you create. As you add sections, the display adjusts according to an underlying content style.

You can choose between 2 content styles:

- The default **above-fold** content style minimizes vertical scrolling by adjusting the size of the first section as you add more.
- The alternate **hero-plus** content style always displays the first section to the full width and height. All other sections appear underneath.

The Default Above-Fold Content Style

In the **above-fold** style, the first panel is the biggest section, but the panel shrinks horizontally and vertically as you add sections. Salespeople don't have to scroll if they have a standard-sized monitor.

Here's how the above fold layout changes depending on the number of sections you specify in the container.

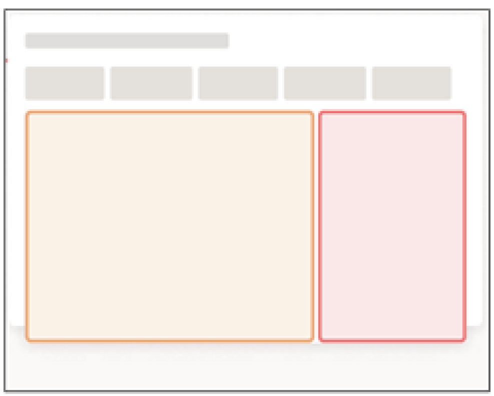
- 1 section

  The section takes up the full width and height of the content region of the tab.

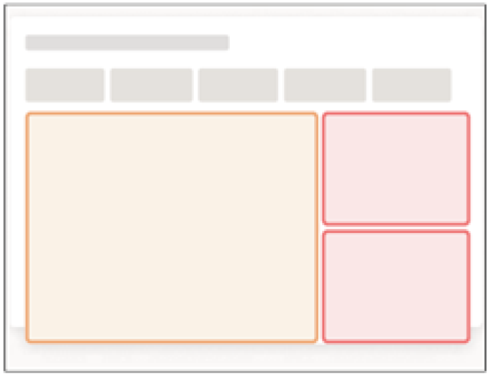
- 2 sections

  The first section takes up two-thirds of the width at full height and the second section takes up the rest.

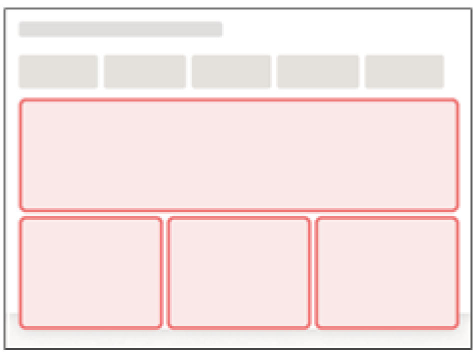
- 3 sections

  The first section takes up two-thirds of the width at full height and the second and third sections split up the remaining third horizontally.

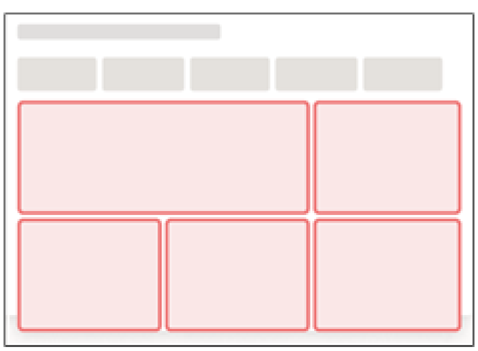
- 4 sections

  The first section takes up the full width of the dashboard but only half of the height. The three other sections split the remaining space below the first section.

- 5 sections

  The first section takes up two-thirds of the width at half the height, The second section follows with one third of the width and at half the height. The three remaining sections split the remaining space below the first section.

The Alternate Hero-plus Layout

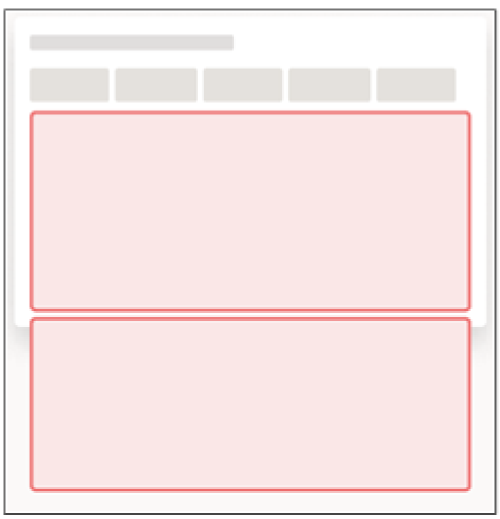
The **hero-plus** layout displays the first section, the most important information, over the full width and height of the available space. As you add sections, they appear below. Salespeople may have to scroll down to see them.

- 1 section

  The first section takes up the full width and height of the space available.

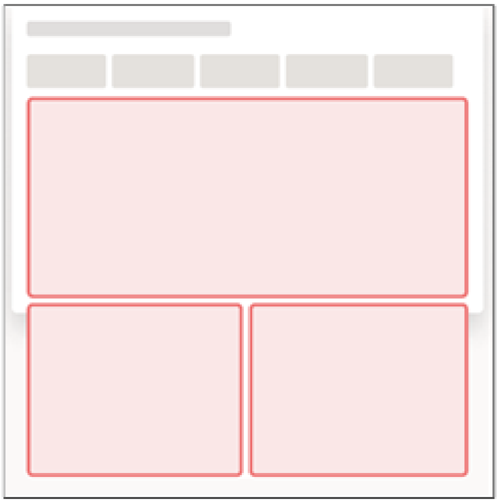
- 2 sections

  The first section takes up the width at full height. The second section displays below.



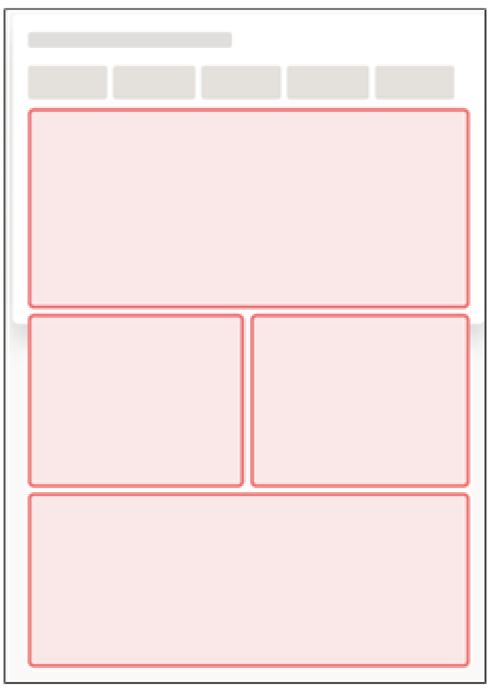
- 3 sections

  The first section takes up the width at full height. The remaining 2 sections display below next to each other, each at half of the width.

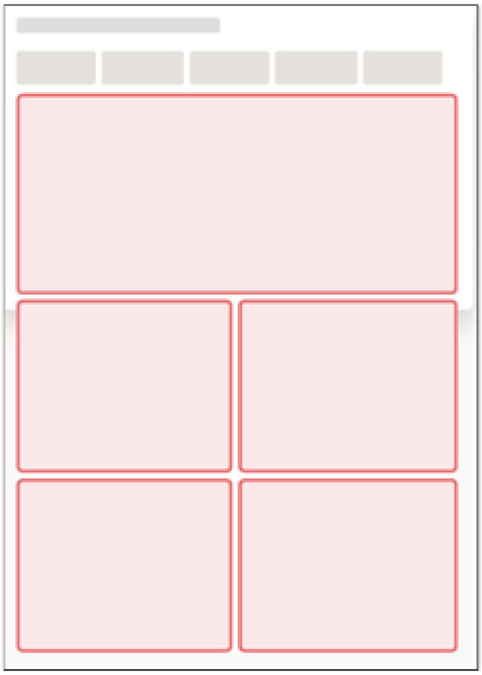
- 4 sections

  The first section takes up the width at full height. The next 2 sections display below, each at half of the width. The fourth section appears at full width below.

- 5 sections

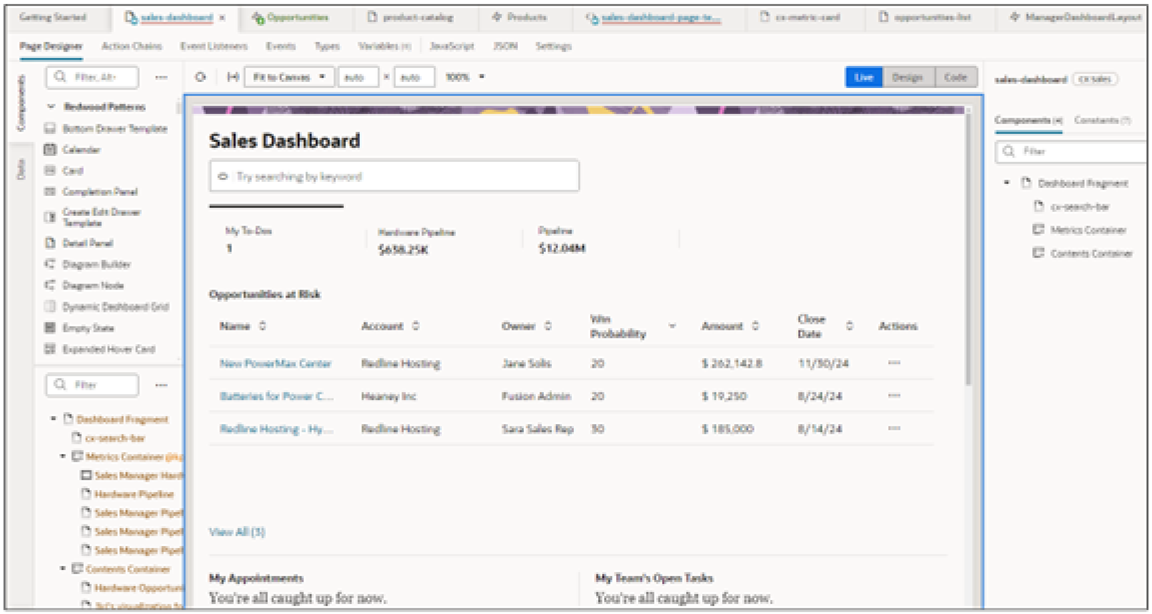
  The first section takes up the full width and height of the dashboard. The rest of the sections appear underneath at half the width.

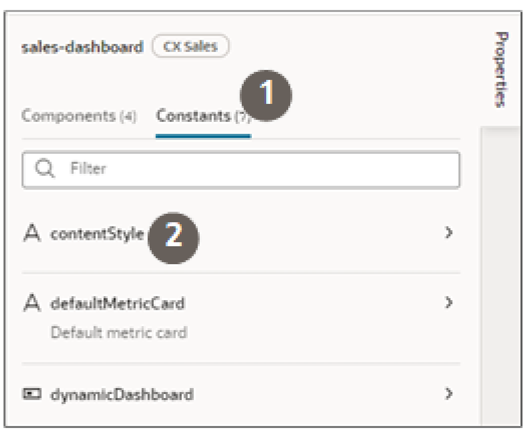
Switch Between the Two Sales Dashboard Content Styles

Here's how to switch to the alternate hero-plus style or back to the default. The hero-plus style, gives priority to the first section in the contents container and displays it over the full width and height of the available space.

1. Edit the Sales Dashboard page in Oracle Visual Builder Studio.

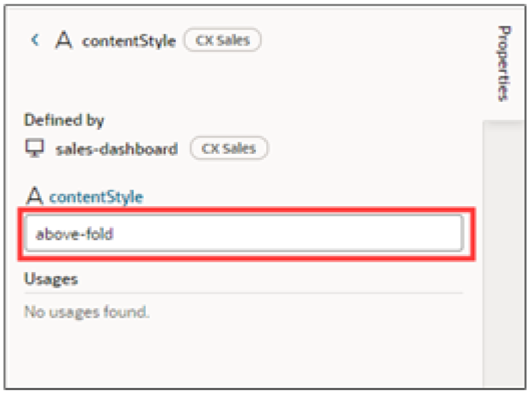


2. In the Properties pane on the right, click **Constants** (the tab highlighted by callout 1 in the following screenshot).



3. Click **Content Style** (callout 2).

4. Switch content styles by entering either **hero-plus** or **above-fold** in the **ContentStyle** field.



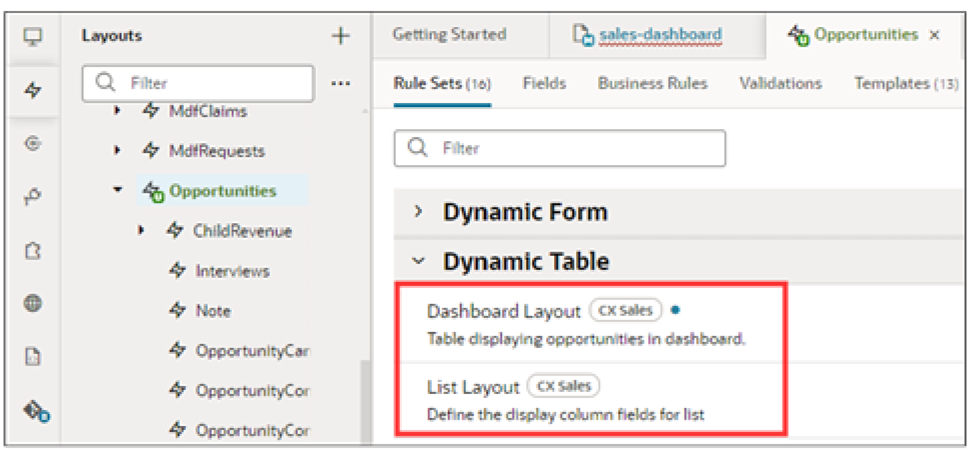
Layouts That Specify What Fields Display in Tables and Lists and How

The contents container specifies the source of information for a table and a list. You use a layout to specify which fields display as columns in the table or a list. The layouts determine which fields display, their order, and their relative widths to other fields.

> **Note:** If you created your table as a visualization, this topic doesn't apply to you. That's because you specify the columns you want to display in the tabular chart visualization itself.

There are separate layouts for different sales objects. For opportunities, the 2 relevant sales dashboard layouts are listed under the **Dynamic Table** heading:

- **Dashboard Layout**, which specifies the columns in tables.
- **List Layout**, which defines the fields in lists.



You can create different layouts for different containers.

To understand how to specify the fields in layouts, see the example provided in *Specify the Columns in a Sales Dashboard Table and Their Widths*. You can edit list layouts in the same way.

Specify the Columns in a Sales Dashboard Table and Their Widths

To specify which columns appear in your sales dashboard table and control their width, you copy and configure the dynamic table Dashboard Layout.

1. In Oracle Visual Builder Studio, click the Layout tab and open the CX Sales node.
2. Scroll down or search for **Opportunities**.
3. Click **Opportunities** to display the opportunity layouts.
4. In the **Dynamic Table** section, click **Dashboard Layout**.
5. Duplicate the **isDefaultDashboardLayout**.
6. Open the copy.
7. Select and drag the fields you want to display from the left pane to the **Select fields to display** pane.
8. You can reorder the columns in your table by using the handles in front of each field.
9. You can adjust the width of each field by clicking on the field. The width controls display as percent of the table length.

   For example, the opportunity **Name** width is set to 19% by default, so the name column will always display at 19% of the total width of the table at the default window size.

   If the user views the dashboard in a smaller window, then any field without a width value automatically adjusts, but the name column will stay the same width.

   - Clearing all of the widths for all the columns causes the table to adjust all of the columns.
   - If you add more columns that the width of the table supports or the percent entries add up to more than 100%, then the table displays a scroll-bar.

> **Note:** Although the **Minimum Width** and **Maximum Width** fields display 33% by default, these aren't actual values. The percentage value is merely a placeholder.

Add Predefined Content to Your Tab

Adding any of the predefined content provided by Oracle to your tab is as simple as clicking the **Add Section** button and making a selection from a list.

1. If the copy of the content container you made already includes 5 sections, then you must remove a section by selecting a row and clicking **Remove Item** (the trash can icon).
2. Click **Add Section**.
3. To add any of the content predefined by Oracle as is, scroll down and select it from the list. The available content includes tables, lists, and 2 visualizations.

   See the *Predefined Content That You Can Add to Your Dashboard Tabs* topic for details.

Predefined Content That You Can Add to Your Dashboard Tabs

Oracle provides you with several predefined tables, lists, and visualizations that you can add to your Sales Dashboard tabs.

You can add these components to any custom Contents Container

Lists

- My Appointments saved search list
- My Favorite Opportunities saved search list
- My List
- My Open Leads saved search list
- My Open Opportunities saved search list
- My Open Tasks saved search list
- My Overdue Tasks saved search list
- My Team's Open Leads saved search list
- My Team's Open Tasks saved search list
- My Team's Overdue Tasks saved search list

Tables

- Deals At Risk saved search table
- My Open Opportunities saved search table
- My Team's Opportunities saved search table
- Tasks Due in the Next 30 Days saved search table

Charts

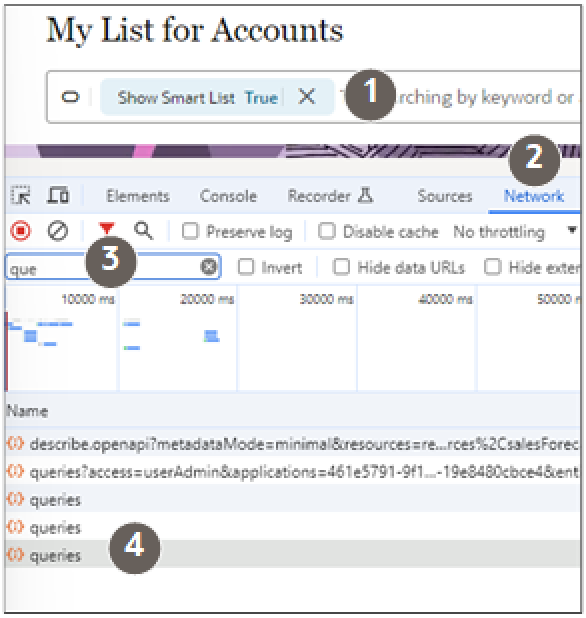
- Lead Amounts by Team Member chart
- My Team's Performance chart

How to Find the UUIDs for Saved Searches and IDs for Its Filters

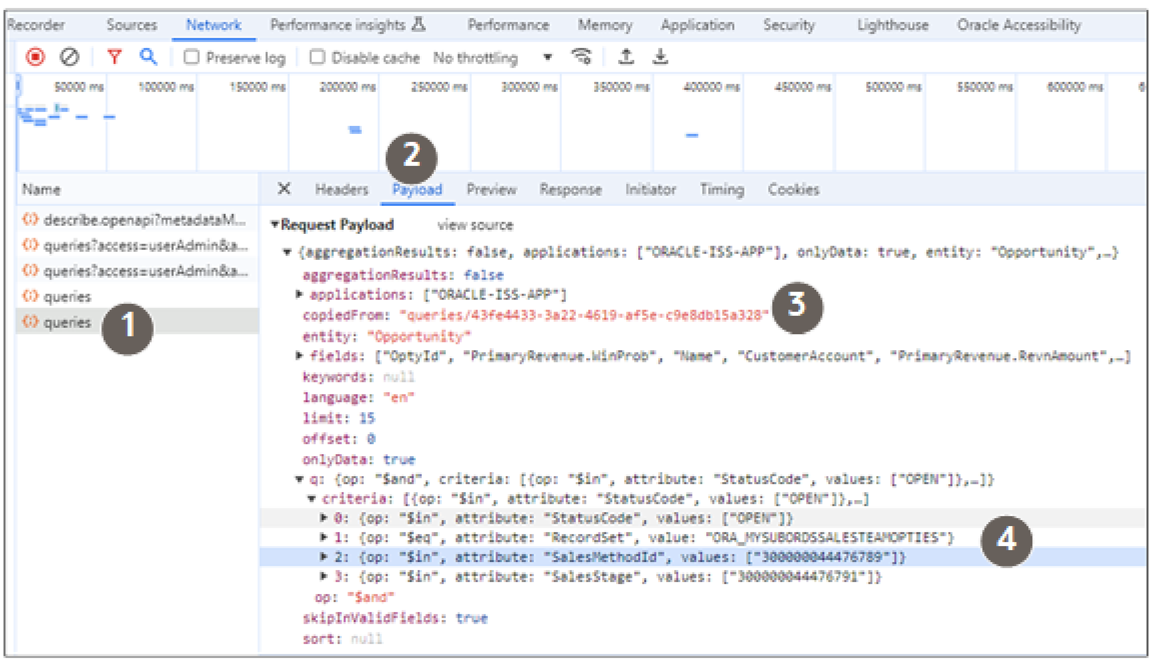
Here's how to find the Universally Unique Identifier (UUID) for a saved search and the various IDs of its query parameters in Oracle Sales in the Redwood User Experience. The IDs are required to reference the saved search and filter parameters in Sales Dashboard layouts, including metric cards and tabular and list views. The IDs are also used in Oracle Transactional Business Intelligence analyses. You'll need an ID to filter by opportunity sales method and sales stage, for example. A UUID is a is a 36-character alphanumeric string that can be used to identify information, such as rows of data within a database table.

Here's how to find the UUID for a saved search:

1. Open the list page of the object in the Chrome browser. For example, if you need the UUID for accounts, for example, click **Accounts** on the home page.
2. Click in the action bar (callout 1 in the following screenshot) and select any saved search other that the one you're looking for.
3. Right-click the page and select **Inspect** from the Chrome browser menu.
4. Click the **Network** tab (callout 2).
5. Reload the browser page.
6. In the action bar, switch to the saved search you want.
7. Click the **Network** tab again.
8. Enter **que** in the filter field to filter out the queries (callout 3).
9. Select the last **queries** entry in the list (callout 4).



10. With the last queries item selected (callout 1 in the following screenshot), click the **Payload** tab (callout 2).
11. The UUID is the string of characters following **queries/** in the **copiedFrom:** line. For example: `934fb7ec-658f-4718-b38c-95f8f5e6c431` (highlighted by callout 3).



12. To view the IDs of the query parameters, open the **q:** list (callout 4). For example, here's a sample sales method ID:`{op: "$in", attribute: "SalesMethodId", values: ["300000044476789"]}`

Values to Use in Code for Standard Components in Tables, Metric Cards, and List Views

Here's a list of resource parameters and values to use in your Sales Dashboard code for the different standard objects.

Opportunities

Resource parameter: `<oj-vb-fragment-param name="resource" value='[[ {"name": "opportunities", "primaryKey": "OptyId", "puid": "OptyNumber", "endpoint": "cx" } ]]'></oj-vb-fragment-param>`:

Name	Primary Key	Puid	Endpoint	Default List Template ID	Default Table Template ID
opportunities	OptyId	OptyNumber	cx	dashboardCardLayout	dashboardLayout

Leads

Name	Primary Key	Puid	Endpoint	Default List Template ID	Default Table Template ID
leads	LeadId	LeadNumber	cx	dashboardCardLayout	dashboardLayout

Value to use for `<oj-vb-fragment-param name="resource" value=`:

```
[[ {"name": "leads", "primaryKey": "LeadId", "puid": "LeadNumber", "endpoint": "cx" } ]]
```

Accounts

Name	Primary Key	Puid	Endpoint	Default List Template ID	Default Table Template ID
accounts	PartyId	PartyNumber	cx	dashboardCardLayout	dashboardLayout

Value to use for `<oj-vb-fragment-param name="resource" value=`:

```
[[ {"name": "accounts", "primaryKey": "PartyId", "puid": "PartyNumber", "endpoint": "cx" } ]]
```

Contacts

Name	Primary Key	Puid	Endpoint	Default List Template ID	Default Table Template ID
contacts	PartyId	PartyNumber	cx	dashboardCardLayout	dashboardLayout

Value to use for <oj-vb-fragment-param name="resource" value=:

```
[[ {"name": "contacts", "primaryKey": "PartyId", "puid": "PartyNumber", "endpoint": "cx" } ]]
```

Tasks

Name	Primary Key	Puid	Endpoint	Default List Template ID	Default Table Template ID
activities	ActivityId	ActivityNumber	cx	taskDashboardCardLay	taskDashboardLayout

Value to use for <oj-vb-fragment-param name="resource" value=:

```
[[ {"name": "activities", "primaryKey": "ActivityId", "puid": "ActivityNumber", "endpoint": "cx" } ]]
```

Appointments

Name	Primary Key	Puid	Endpoint	Default List Template ID	Default Table Template ID
activities	ActivityId	ActivityNumber	cx	appointmentDashboarc	appointmentDashboardLayout

Value to use for <oj-vb-fragment-param name="resource" value=:

```
[[ {"name": "activities", "primaryKey": "ActivityId", "puid": "ActivityNumber", "endpoint": "cx" } ]]
```

Custom Objects

Name	Primary Key	Puid	Endpoint
<object>_c	Id	Id	cx-custom

Value to use for <oj-vb-fragment-param name="resource" value=:

```
[[ {"name": "<object>_c", "primaryKey": "Id", "puid": "Id", "endpoint": "cx-custom" } ]]
```

Enable Filtering of Sales Dashboard Information

You can expose filters in the Sales Dashboard that salespeople can use to narrow down the scope of the sales information they view on the sales dashboard tabs. By applying filters, they can focus on specified period of time or on select team members.

This section explains how the search bar works and how you can expose the filters.

ORACLE

The Dual Functionality of Sales Dashboard Search

The Sales Dashboard search box serves two different and unrelated functions: keyword searches and the filtering of the reports in the sales dashboard itself.

Salespeople use the search box to:

- Search by keyword through all the sales information they can access, including activities, emails, leads, opportunities, accounts, and other business objects.
- Filter the information in the active tab of the sales dashboard. For example, they can display data for a narrower time period or compare the performance of one manager to another.

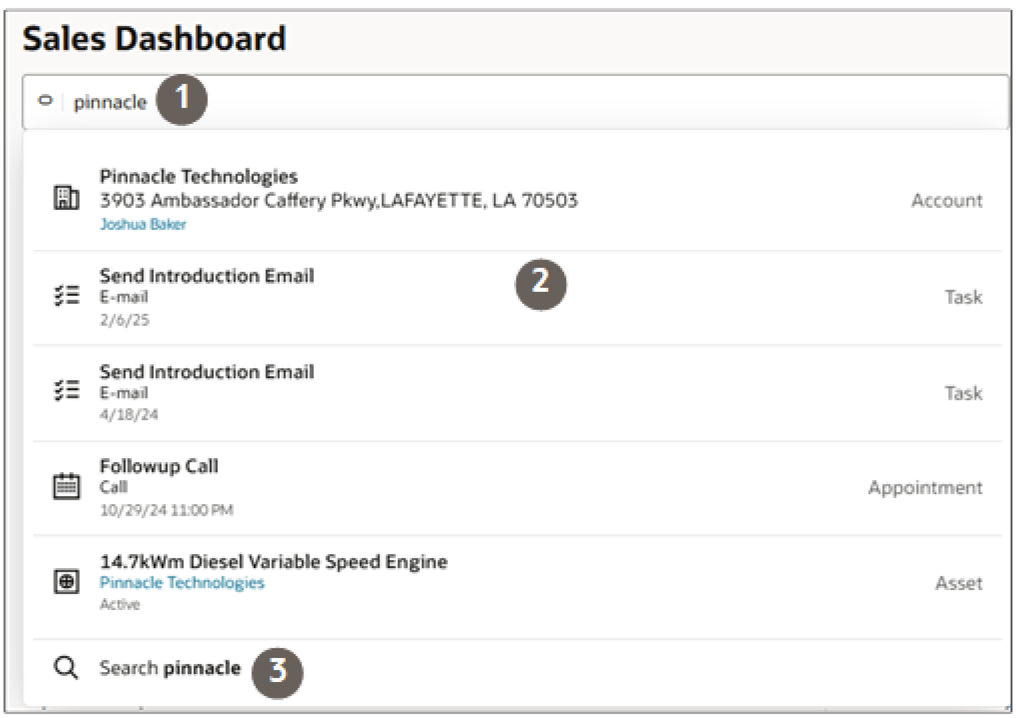
Keyword Search

Here's how to search by keyword:

1. Enter the keyword in the search bar.
2. If you don't see the item you're looking for in the list, click **Search** to view all the results, filter them, and take actions.

Keyword Search

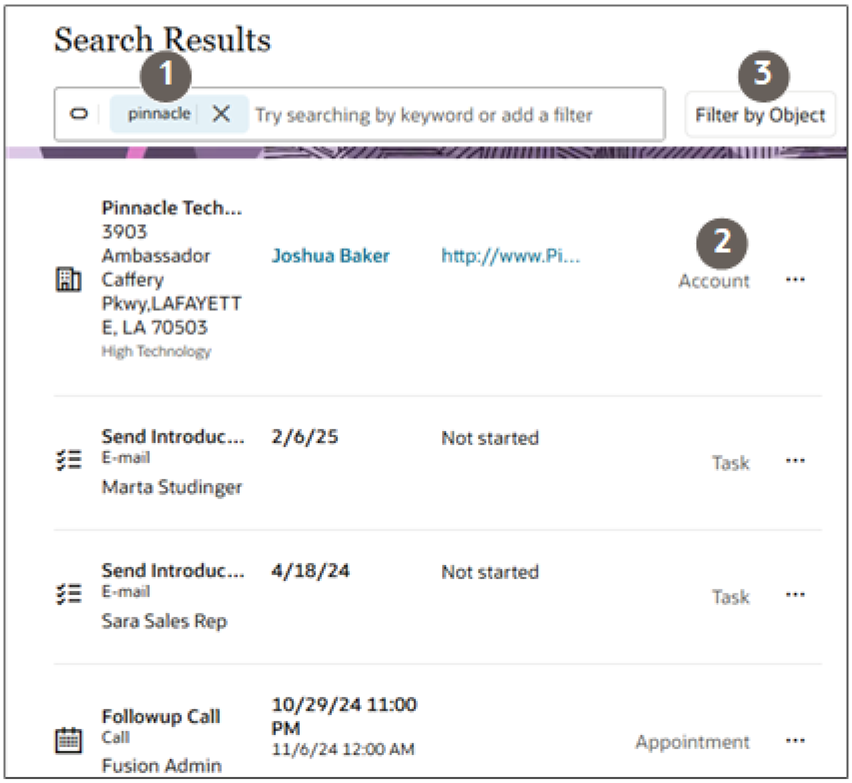
Callout	Description
1	Enter keyword.
2	View list of most often-accessed items matching the keyword.
3	Click **Search** to view all results and take actions on them.



3. In the Search results page, you can add different search terms and filter by object.

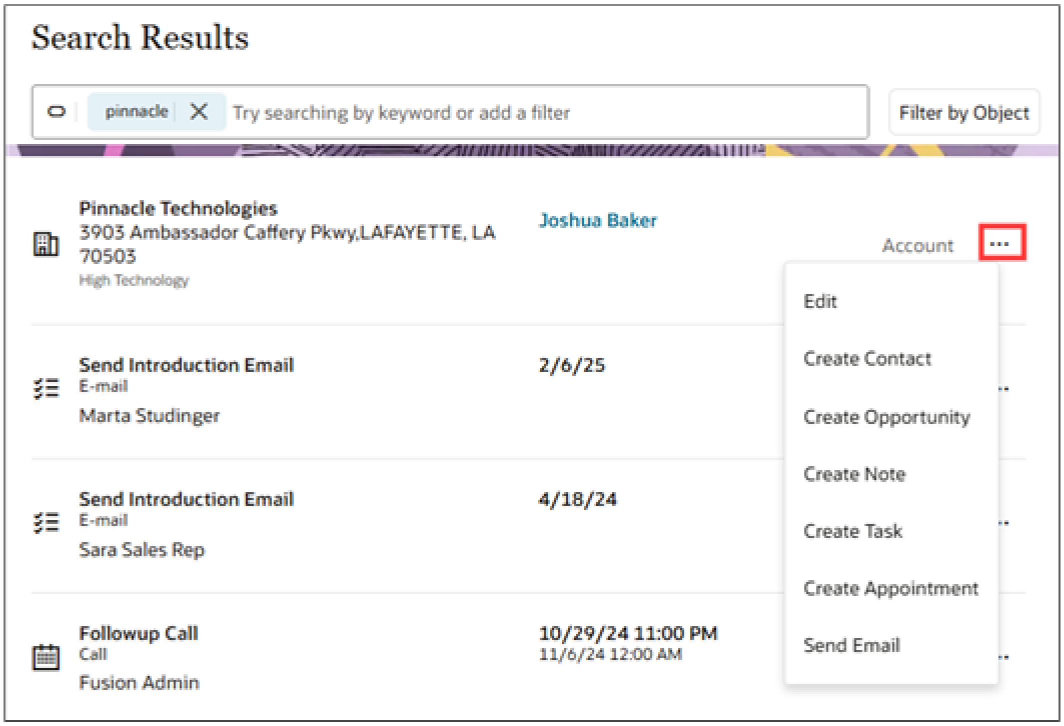
Key Features of the Search Results Page

Callout	Description
1	You can update the initial keyword.
2	Search results are classified by object.
3	You can filter the results by object.



4. You can take actions directly on the search results by clicking the three dots **Actions** icon highlighted in the screenshot.

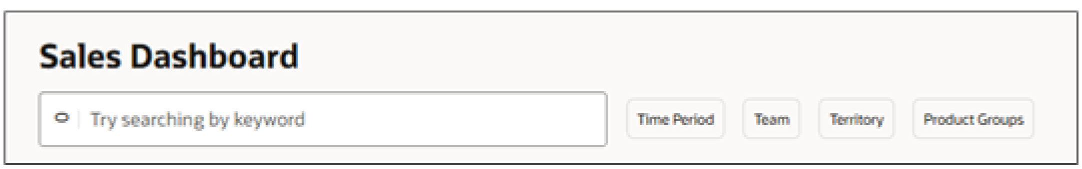
   For example, you can edit an account record, and create contacts, opportunities, notes, tasks, appointment for the account. And you can and you can send an email.



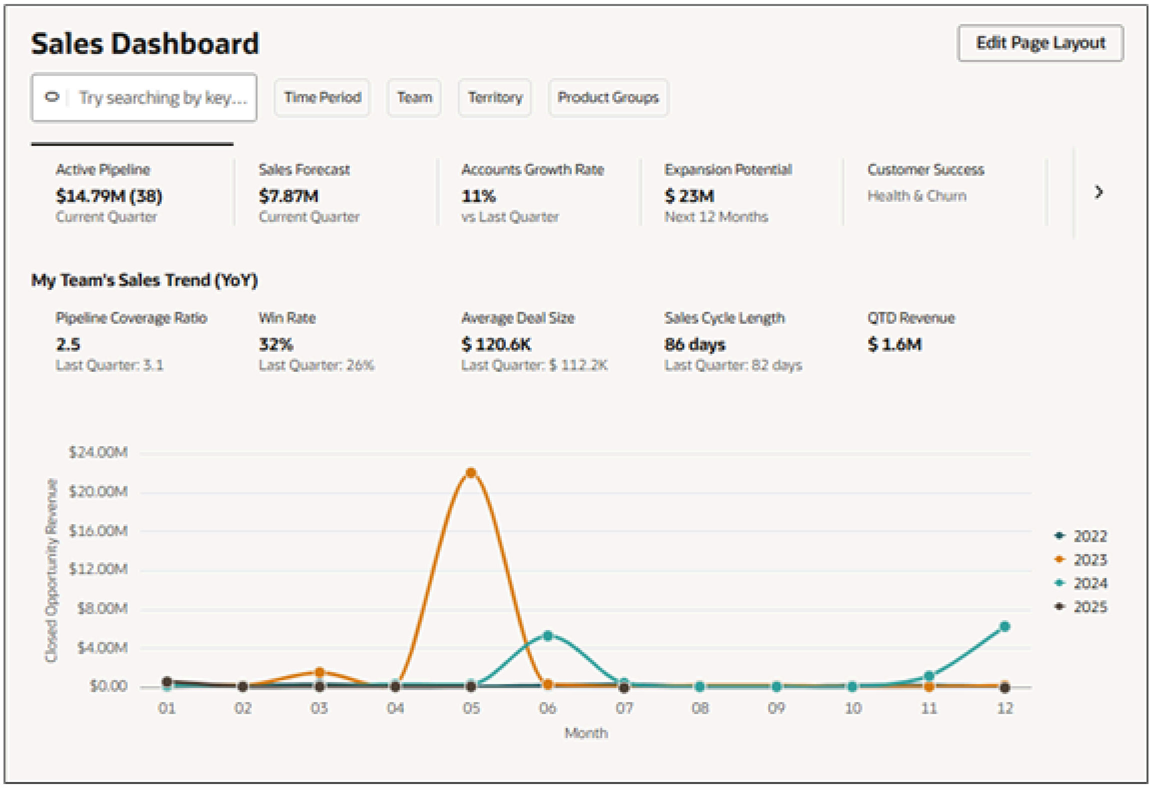
Filter the Sales Dashboard Content

Salespeople can filter the contents of the dashboard to focus on a subset of the information.

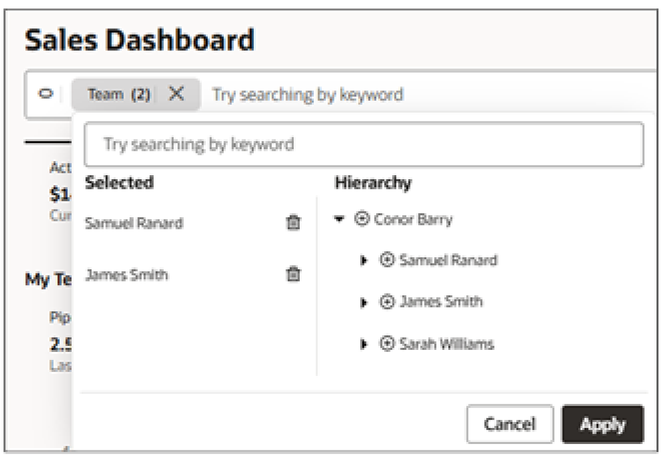
Four filters are available: Time Period, Team, Territory, and Product Groups. The filters display as buttons and are also available for selection in the search box itself.



Here's a screenshot of a dashboard showing the active pipeline for the whole team, both as a metric and a visualization.

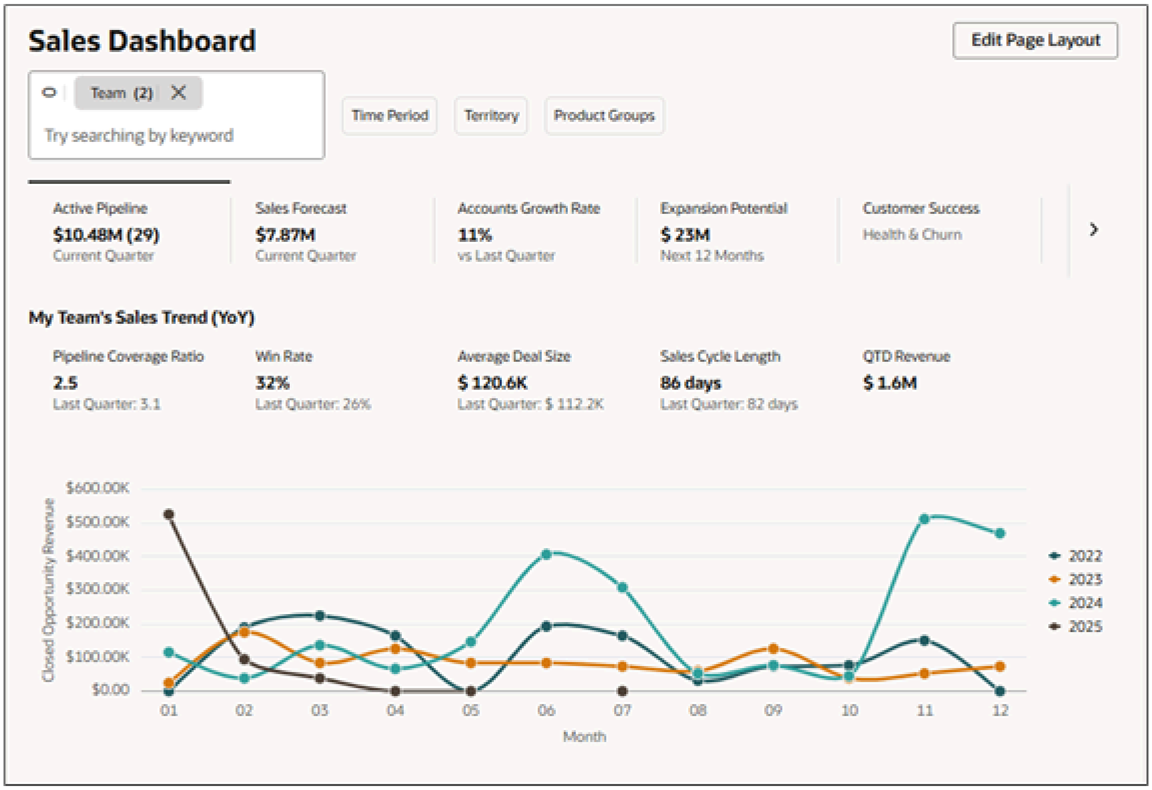


You can view the totals of the active pipeline for just 2 managers by clicking the **Team** filter and selecting the managers.



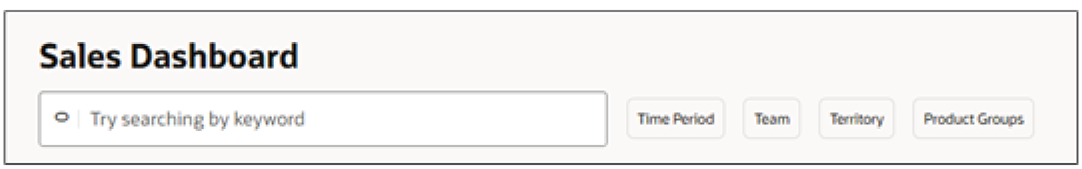
Both the metric and the visualization now display data for the 2 managers.

Note: If you're creating visualizations from OTBI analyses, you must make sure that those analyses include prompted criteria for the filters you enable.



Enable Filtering of Information Displayed in the Sales Dashboard

You can enable 4 filters that salespeople can use to filter the information displayed in the sales dashboard: Time Period, Teams, Territory, and Product Groups. When you enable the filters, they're available in all the sales dashboards in your organization.

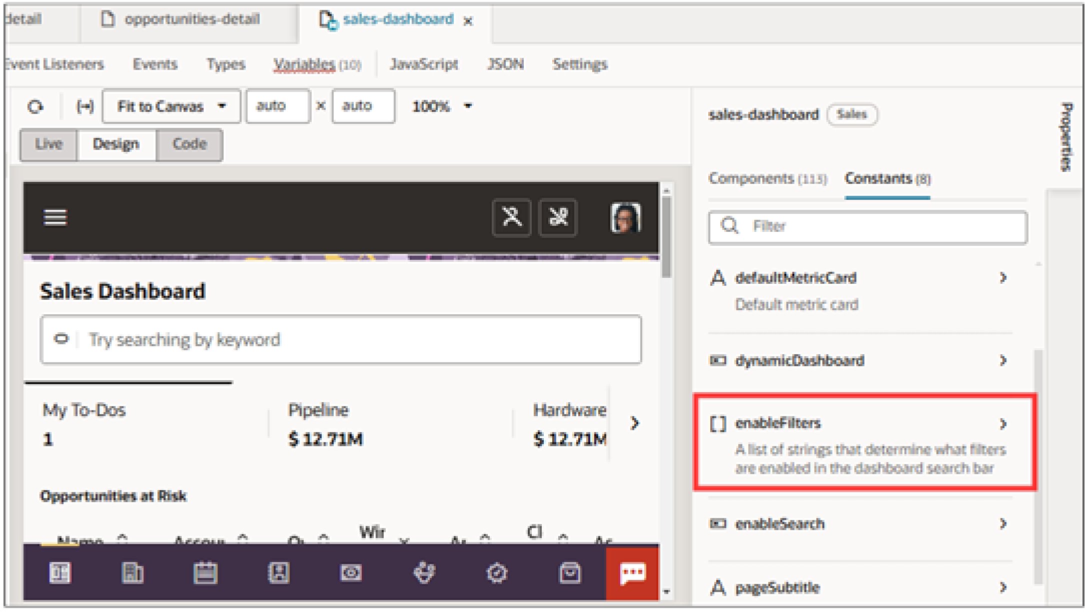


Here's how to display the global filters:

Enable the Filters Using VB Studio

1. Open the custom Sales Dashboard where you're adding the filters.
2. Click **Settings and Actions** > **Edit Page in Visual Builder Studio**.
3. In VB Studio, select the **App UIs** tab.
4. On the **Page Designer** subtab, click **Components**.
5. On the **Properties** tab on the right, click **Constants**.

6. Click **enableFilters**.

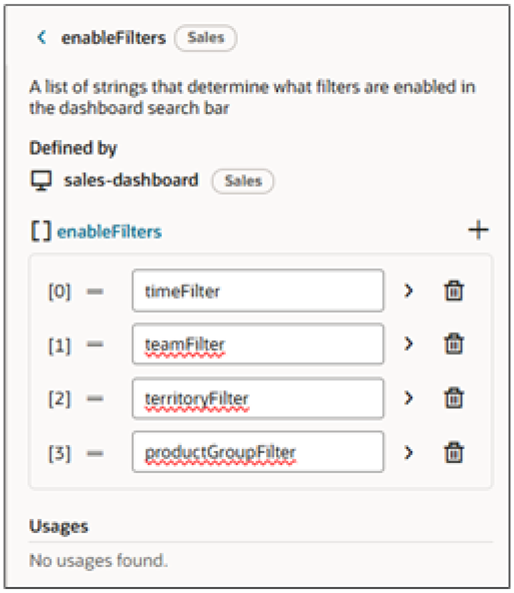


7. Use the **Add Item** (plus sign) icon to add rows for each filter you want to display. You can make the filters available to all or restrict them to specific job roles. For example, only managers may need to filter by teams and territories.

   - Enter the following code to enable filters to all users:

Filter	Code to Enable to All Users
Time Period	`timeFilter`
Team	`teamFilter`
Territory	`territoryFilter`

Filter	Code to Enable to All Users
Product Groups	`productGroupFilter`



- To restrict a filter to specific job roles, enter code similar to the following. The sample code below restricts the metric you specify in **yourMetric** to the sales manager roles provided by Oracle.

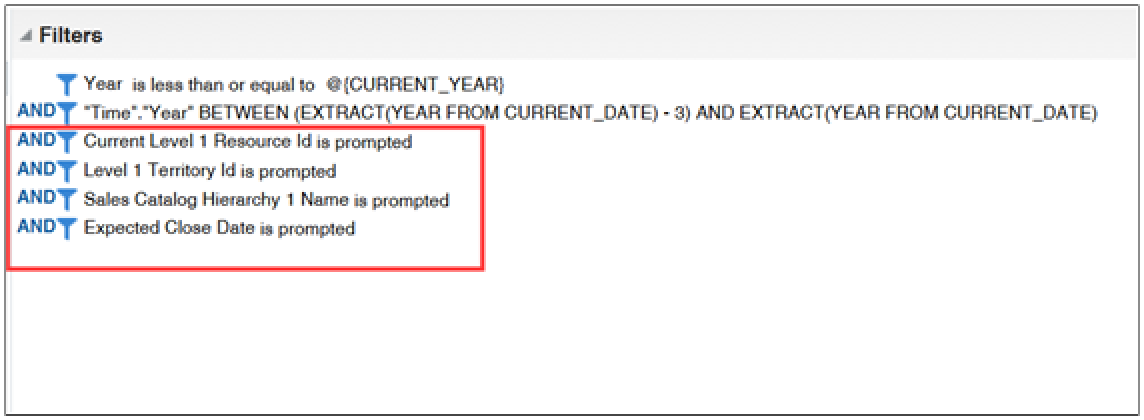
```
[[ $application.user.roles.includes('ORA_ZBS_INSIDE_SALES_MANAGER_JOB') ||
 $application.user.roles.includes('ORA_ZBS_SALES_MANAGER_JOB') ? 'mymetric' : 'yourMetric' ]]
```

Enable Filters in the OTBI Analyses

You must include "is prompted" column filters in your analyses that match the filters you enable in the dashboard. The following table lists the columns for opportunity pipeline visualizations.

Filters to Add in OTBI to Enable Sales Dashboard Filtering

Filter	Folder	Column	Operator
Team	**Sales - CRM Pipeline** > **Employee** > **Sales Resource Hierarchy/**	**Current Level 1 Resource id**	is prompted
Territory	**Sales - CRM Pipeline** > **Territory** > **Territory Hierarchy**	**Level 1 Territory id**	is prompted
Time	**Sales - CRM Pipeline** > **Opportunity**	**Expected Close Date**	is prompted
Product Group	**Sales - CRM Pipeline** > **Product**	**Sales Catalog Hierarchy 1 Name**	is prompted

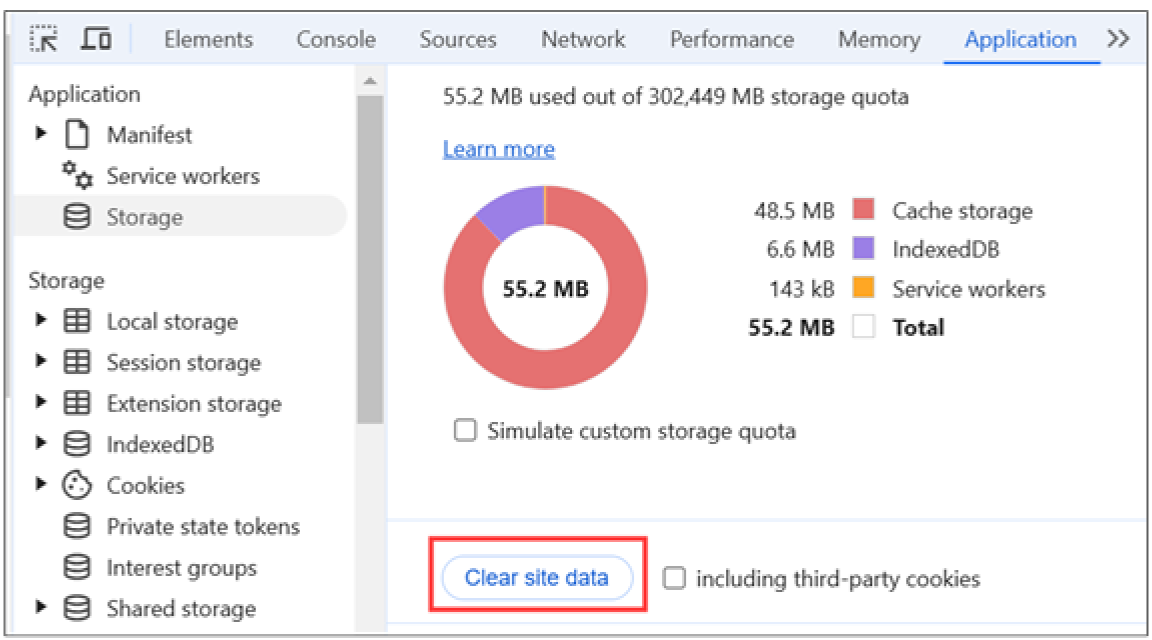


What do I do if the filters I apply aren't reflected in the sales dashboard?

If the filters aren't reflected in the sales dashboard, try the following:

- Refresh the browser.
- Clear site data in your browser:

  a. In the Chrome browser, right-click on the sales dashboard page.
  b. Select **Inspect**.
  c. Click **Applications** subtab.
  d. Click **Clear site data**.

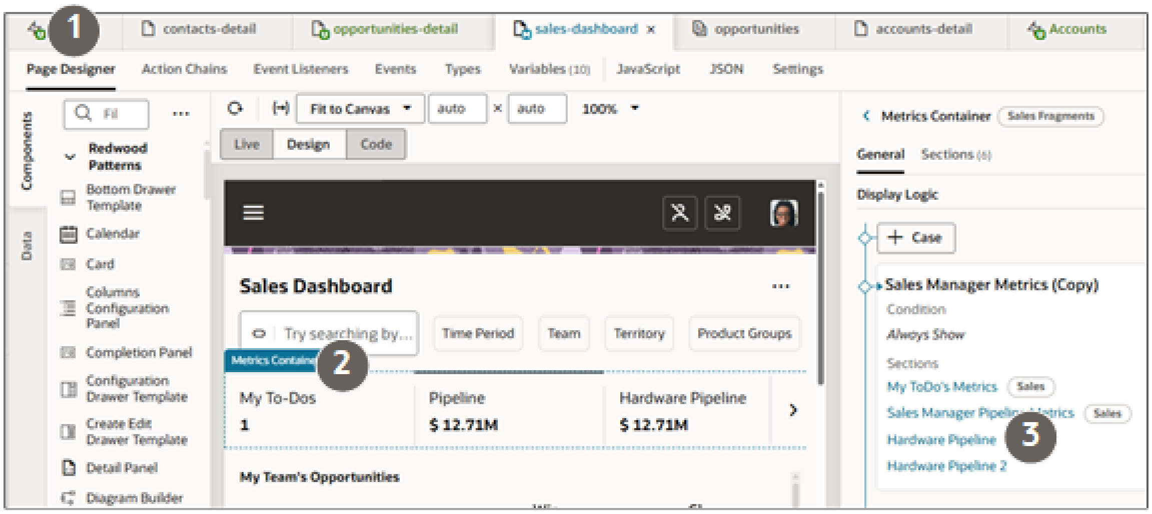


- Check to make sure the OTBI analysis includes "is prompted" filters. See *Enable Filters in the OTBI Analyses*.

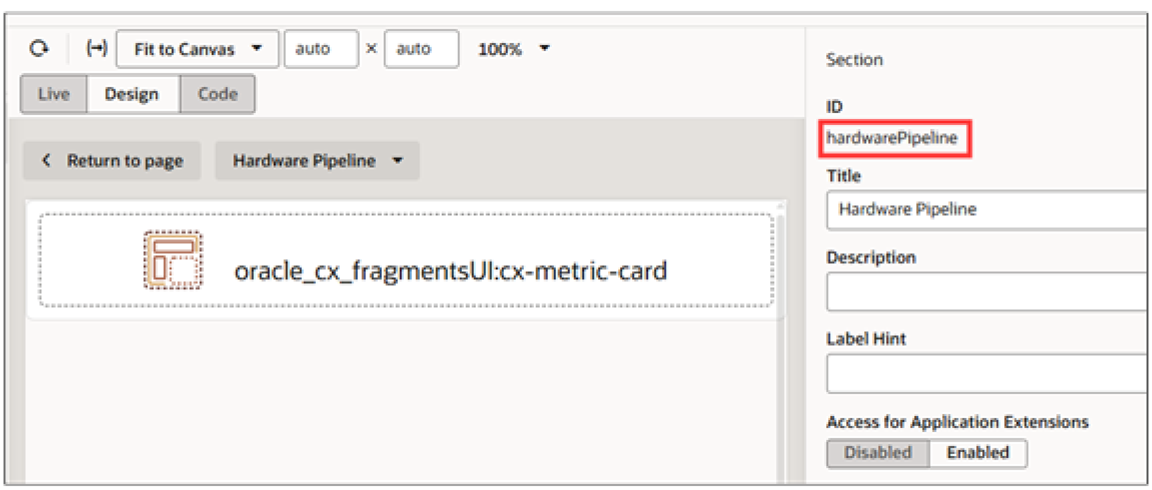
Specify which Dashboard Tab Displays by Default

You can specify which sales dashboard tab is selected and displays when a user opens the Sales Dashboard by setting the constant **defaultMetricCard** in Oracle Visual Builder Studio.

1. Open the Sales Dashboard in VB Studio.
2. Obtain the ID of the metric card:
   a. On the **sales-dashboard** tab, click the **Page Designer** subtab.
   b. In the Design mode, click the **Metrics Container** section.
   c. Drill down into the metric card (section) in the Display Logic pane.

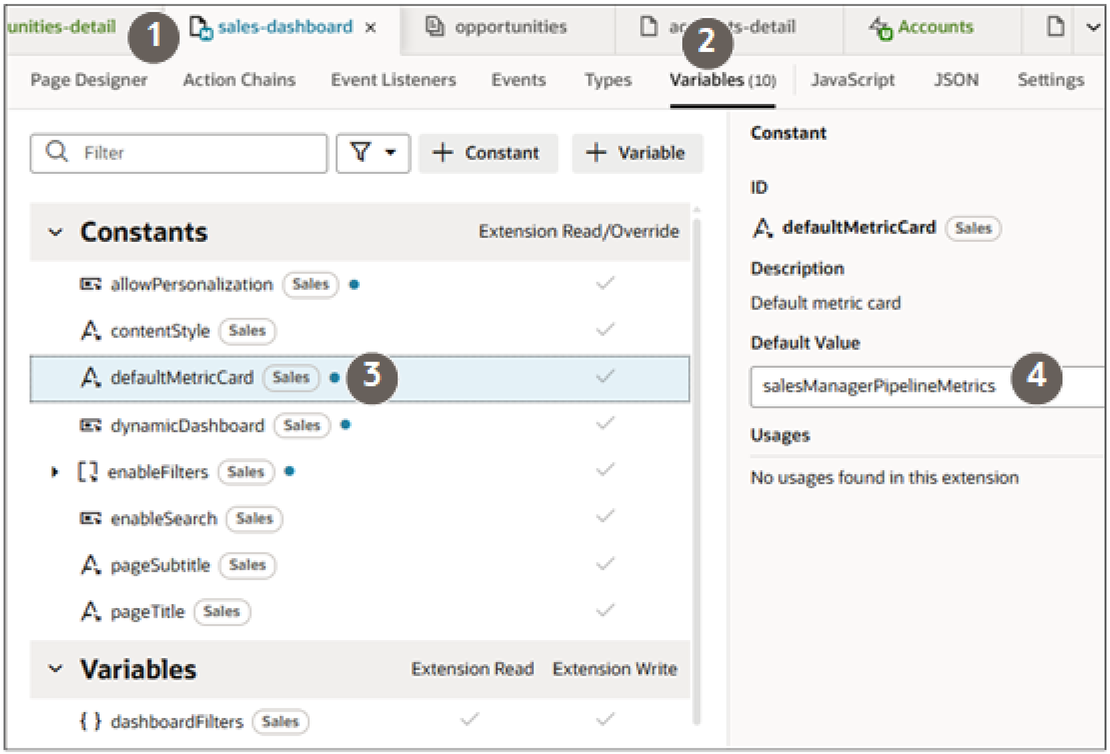


   d. copy the ID.



3. On the **sales-dashboard** tab, click the **Variables** subtab.
4. Click the **defaultMetricCard** constant.

5. In the **Default Value** field, enter the ID of the metric card for the tab you want to show as the default.

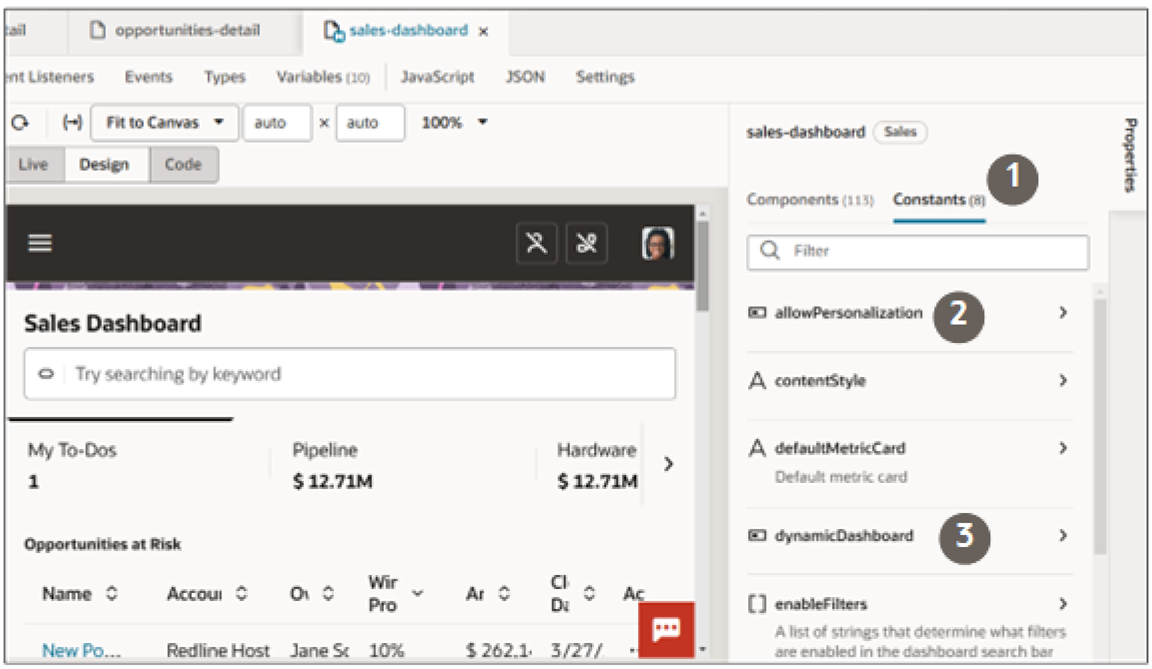


5 Enable and Use Sales Dashboard Personalization

Enable Sales Dashboard Personalization

Here's how to make it possible for users to configure their own sales dashboard versions using the content provided by administrators.

1. Open the Sales Dashboard version where you want to enable editing.
2. Click **Settings and Actions** > **Edit Page in Visual Builder Studio**.
3. In VB Studio, select the **App UIs** tab.
4. On the **Page Designer** subtab, click **Components**.
5. On the **Properties** tab on the right, click **Constants**.



6. Click **allowPersonalization** and set to **true**.
7. Click **dynamicDashboard** and set to **true**.
8. You can now click **Preview** and use the **Edit Page Layout** button to personalize the sales dashboard.

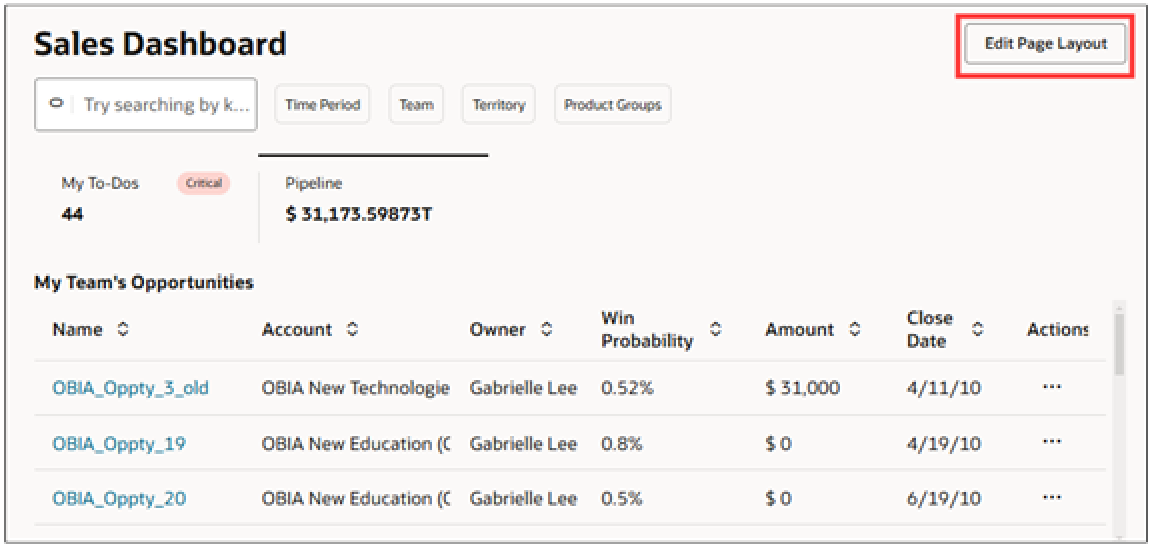
9. Publish to enable the personalization.

Personalize Your Sales Dashboard

You can add, remove, and rearrange content to create your own personal dashboard using the content provided by administrators.

When you personalize the sales dashboard, that personalization is available only to you. You can't share it.

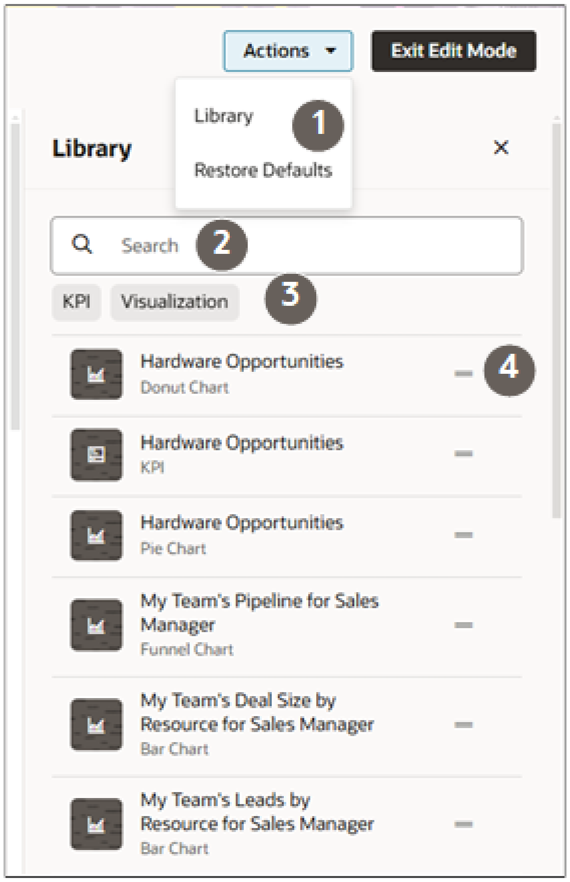
Click the **Edit Page Layout** button to open the editing mode.



Use the **Library** pane to add content. To move an remover items, hover over them and use the controls.

Library Pane Functions

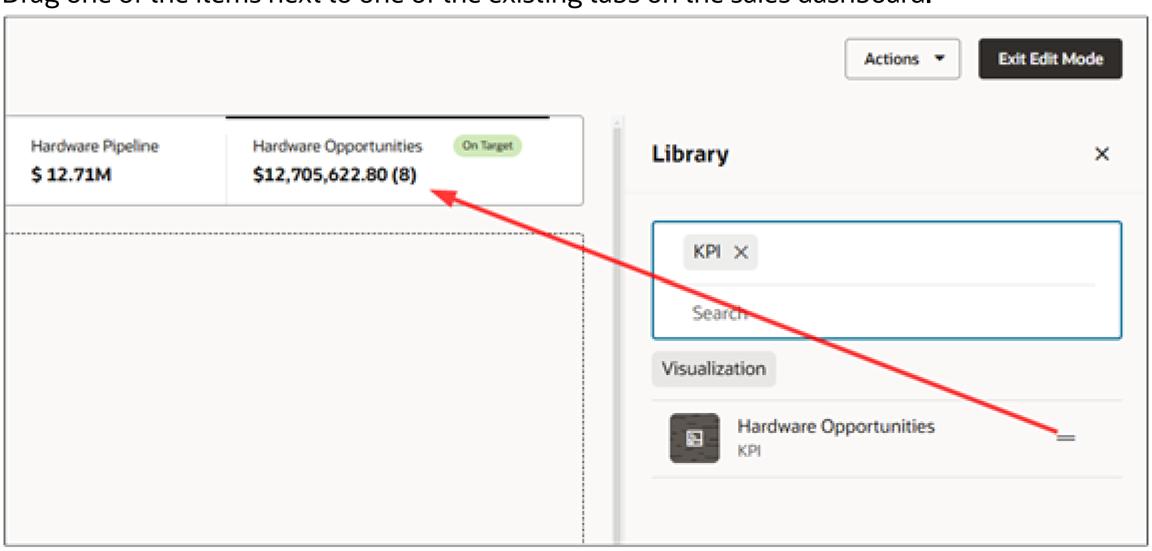
Callout	Feature Description
1	Use the **Actions** menu to restore the original dashboard.
2	Search for content.
3	The **KPI** filter displays the content available to create the top of a tab. The **Visualizations** filter lists the content you can add to the tab itself.
4	Drag items into place using the handles.



Create Tabs

You can create your own tabs. Here's how:

1. In the **Library** pane, click **KPI** to view the available KPIs that you can use for the top of your tab.
2. Drag one of the items next to one of the existing tabs on the sales dashboard.



3. You're now ready to add content.

Add Content to an Existing Tab

You can have maximum of 5 items in a tab. You can add:

- Charts and tables created by administrators in the Visualization Configuration tool.
- Express Reports that you've created and those shared with you by an administrator.

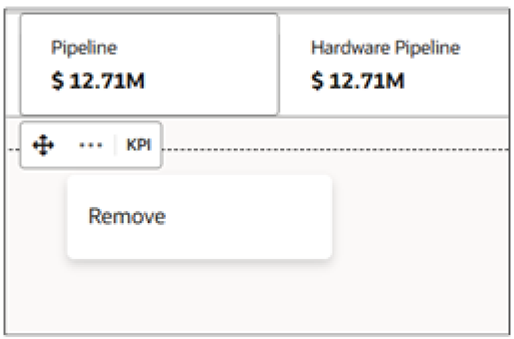
Note: The library includes all available visualizations and express reports. Some may not be applicable for your role in the organization. You may not have permission to view the data in some charts, for example.

Here's how to add content:

1. In the **Library** tab, click the **Visualizations** button.
2. Search for what you want to add in the library.
3. Use the handles to drag the item into place on the dashboard.

Move and Remove Content

- To move an item, hover over it and drag using the handle.
- To remove an item, hover over it and click **Actions** > **Remove**



Restoring the Default Dashboard

You can restore the original dashboard, by clicking **Actions** > **Restore Defaults** (the button at the top of the page).

6 Visualization FAQs

What do I do if a user doesn't see Visualization Configuration in Application Composer?

Make sure that the user is assigned the following job roles provided by Oracle:

- ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB
- ORA_ZBS_SALES_ADMINISTRATOR_JOB

What do I do when Adaptive Search isn't available as a source type for creating visualization?

Check to make sure that Oracle Sales in the Redwood User Experience is enabled. Both the Digital Sales Next Gen UI and Sales in the Redwood UX features in the Sales offering must be selected.

If the option isn't visible even after enabling these features, then clear the browser cache.

Related Topics

- How can I enable Oracle Sales in the Redwood User Experience?

What do I do when there's no data in a visualization?

Check to make sure that you've the same data access as the intended user of the visualization. For a saved search visualization, check the list page. For an Oracle Transactional Business Intelligence (OTBI) analysis, check the results in OTBI.

What do I do when I get the error "Select values for all the required fields to generate the visualization"?

Make sure that you select all the values required for the visualization. For example, you must select a value for the x-axis if you're displaying a bar chart. If all the values are selected, check if the column is a hierarchy column. Oracle Business Intelligence hierarchical data can't be used in visualizations.

What do I do when I get the warning that the analysis has more columns than selected?

Here's what to do when you see the warning that your analysis: "...**has more columns than selected. Remove the unused columns from the analysis else visualization may not show correct data.**"

The attributes that you use in your visualization must use all of the columns in your Oracle Transactional Business Intelligence (OTBI) analysis. If you can't use all of the columns in the visualization, go back to the analysis and remove the unused columns.

For example, a bar chart can support 2 dimensions, 1 metric, and you can have 1 filter attribute. If a report has more than 4 columns (3 dimensions and 1 fact), then you'll see this warning.

Visualizations can't show up correctly if the data in your analysis is aggregated or calculated with more columns than chart can support.

Why does my visualization show a value higher than 100 percent?

Formatting a number in the visualization merely adds a percent symbol (%) to the visualization display. No calculation is performed. You must perform calculations at the source of your visualization data. The same is true for currency symbols.

Why isn't the Meter Gauge and Tabular Chart available for Adaptive Search visualizations?

The Meter Gauge and Tabular Chart visualization types are available only for Oracle Transactional Business Intelligence (OTBI) analyses.

What visualizations are available for Oracle CX Sales Mobile?

Here are the visualizations that are available in for CX Sales Mobile application:

- Pie
- Donut

- Bar (Stacked Bar)
- Area Charts
- Funnel
- Scatter
- Bubble
- Combo- Partially

Not supported are:

- Meter
- Metric
- Tabular
- Line
- Combo (New updates to the charts)

How can I change the sort order in a table I created as a visualization from an OTBI analysis?

You can change the sort order in the Oracle Transactional Business Intelligence (OTBI) analysis. Starting with update 25A, you can also change the sort order in the visualization itself.

How many rows of data does a table visualization show?

The table you created as a visualization displays the first 100 rows of data using the sort order in the underlying Oracle Transactional Business Intelligence (OTBI) analysis. Users can click the View All option on the table to display up to 2000 rows.

What do I do when an OTBI report isn't listed while creating a visualization?

Check the following:

- Can you access the report in Oracle Transactional Business Intelligence (OTBI)?
- Is the correct top-level folder selected? If not, change the folder using the **Select Catalog Folder** button.
- Have you just created the analysis? New OTBI analyses take a while to appear in the visualization configuration tool.

What do I do when the saved visualization configuration doesn't render the chart?

Check if any columns in the Oracle Transactional Business Intelligence (OTBI) analysis were added or deleted. If they were, then you must add the changed columns again because the visualization relies on column IDs from the analysis.

What do I do if tick mark labels repeat on a chart?

If tick marks repeat on a chart, you can switch the **Number Format** from **Currency** to **Decimal** to fix the issue. This is a known issue that's in the process of being fixed.

How come I don't see all the rows of my data in a table I created as a visualization?

When you create a tabular visualization of the data from an Oracle Transactional Business Intelligence analysis, the table shows a maximum of 100 rows. Users can click **View All** to see all of the records.

How come that, in the Sales Dashboard, I can't sort the table created as a visualization?

You can't sort a table that's created as a visualization in the Sales Dashboard UI. Instead, you must change the sort order in the Oracle Transactional Business Intelligence (OTBI) analysis itself.

What's the cause of the error "source doesn't have enough value to generate a visualization"?

Check your report to see if you have the required number of attributes in the Oracle Transactional Business Intelligence analytic that's the source of your data. For Metric Charts, you must include at least 1 Fact column. For other charts, you must include at least 1 metric and dimension.

What's the likely cause of the message that tells me to reduce the number of metrics for visualization?

There's a limit to the number of metrics that can be set to the "not null" value. The message "Reduce the number of metrics for this visualization. You've exceeded the maximum number." appears when this maximum number is exceeded.

What does it mean when I get the error "We couldn't create a BI session"?

The error indicates that the application is unable to connect to Oracle Business Intelligence (BI) Answers. Check to make sure that BI is running and that you can connect to BI Answers.

7 Sales Dashboard Configuration FAQs

How come I don't see the Sales Dashboard icon?

The Sales Dashboard is only visible to job roles with the privilege ZBS_VIEW_SALES_DASHBOARD_PRIV.

The privilege is available in the following job roles provided by Oracle:

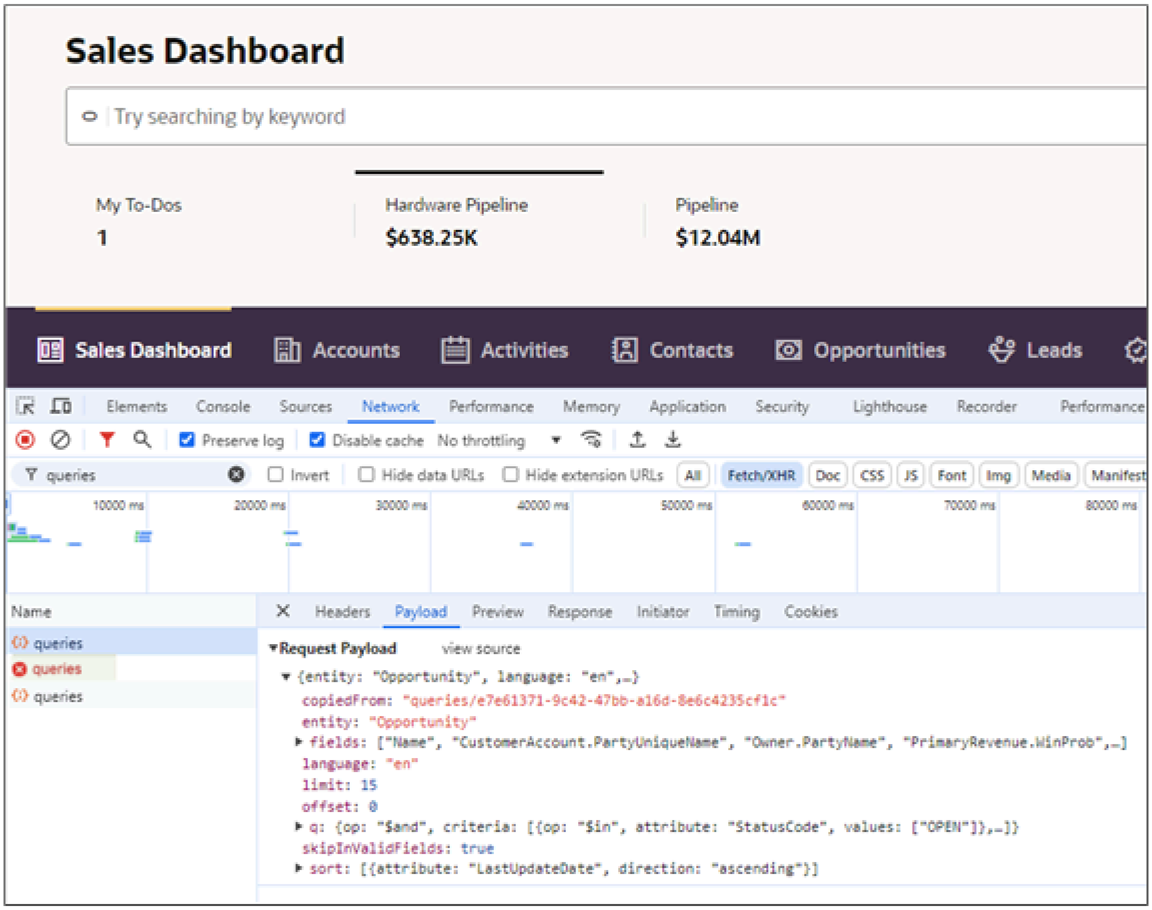
- ORA_ZBS_SALES_REPRESENTATIVE_JOB
- ORA_ZBS_SALES_MANAGER_JOB
- ORA_ZBS_SALES_VP_JOB

What do I do if a dashboard table or list doesn't show any data from a saved search?

On the list page, check if the saved search is visible to the user. Saved searches can be specific to individual job roles or hidden by administrators.

If the saved search shows up correctly on the list page, then you can troubleshoot using your browser to see if any of the queries have failed:

1. In the Chrome browser, open the sales dashboard from the home page or preview it from Oracle Visual Builder Studio.
2. Right-click and select **Inspect**.
3. Click **Network**.
4. Select the **Preserve log**, **Disable cache**, and **Fetch/XHR** options.
5. Enter **queries** in the **Filter** field.
6. Any query that's failed is labeled with an X in the **Name** column.
7. To troubleshoot, select the query and click **Payload**.



What do I do if the dashboard is blank after I configured it?

Inspect the page in the browser for any errors.

1. With the Sales Dashboard open, right-click and select **Inspect**.
2. Go to the **Console** tab and look for any errors.
3. Check the **Network** tab for any errors in network calls (highlighted in red).
4. If the cause of the error isn't clear, then capture the errors and contact Oracle support.

Why is a field, such as currency or date, not showing up in the correct format?

You must add a template to format a field, including the forward slash character (/). Templates include the following: `/currencyTemplate`, `/dateTemplate`, `/singleSelectFCLTemplate`, `/emailTemplate`, `/phoneTemplate`, `/hyperlinkTemplate`, `/percentageTemplate`

How do I troubleshoot saved search data that's not showing up?

If the saved search with the issue is one of the saved searches provided by Oracle, then check if any of the saved searches have been hidden and enable them again. If it's a custom saved search, then check that the correct saved search is being used and ensure that the user has access to the data.

Use this troubleshooting method for any of these types of errors:

- An "Invalid Identifier for Saved Search" error on a Metric Card
- A dashboard table that shows: "No data to display."
- A dashboard panel that shows the error: "Not able to fetch query definition"