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?

F90641-11

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

How to Use this Playbook

This playbook provides you the steps you need to configure sales dashboards for your organization. You can configure any number of sales dashboards for different audiences with the content provided by Oracle and your own content.

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. Create charts for display in your dashboard as described in the *Visualization Configuration* chapter.
4. To configure your dashboards, follow the order of steps in the setup overview provided in the topic: *Overview of Sales Dashboard Configuration Steps*.

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.

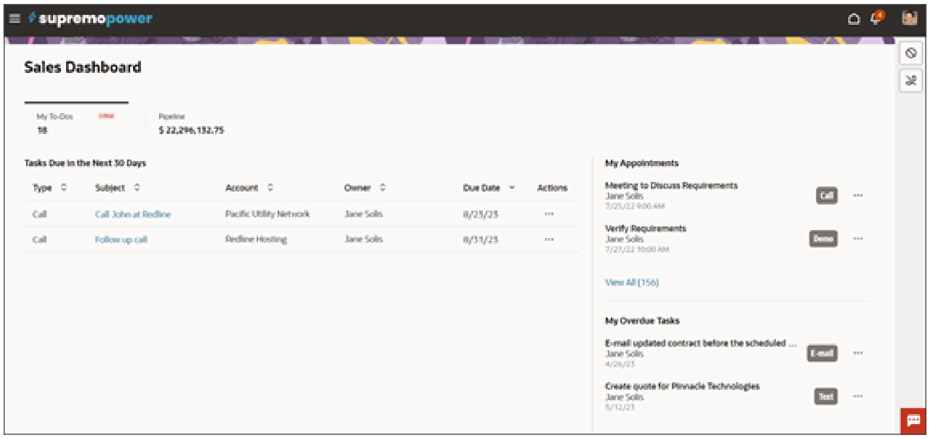
From the dashboard, a salesperson can drill-down into individual items, view all items by clicking View All, and take actions using the Actions menu.

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-Do**



Sales Dashboard

My To-Dos 18

Pipeline $ 22,296,132.75

Tasks Due in the Next 30 Days

Type	Subject	Account	Owner	Due Date	Actions
Call	Call John at Redline	Pacific Utility Network	Jane Solis	8/23/23	...
Call	Follow up call	Redline Hosting	Jane Solis	8/31/23	...

My Appointments

Meeting to Discuss Requirements
Jane Solis
7/25/22 9:00 AM
Call

Verify Requirements
Jane Solis
7/27/22 10:00 AM
Demo

View All (156)

My Overdue Tasks

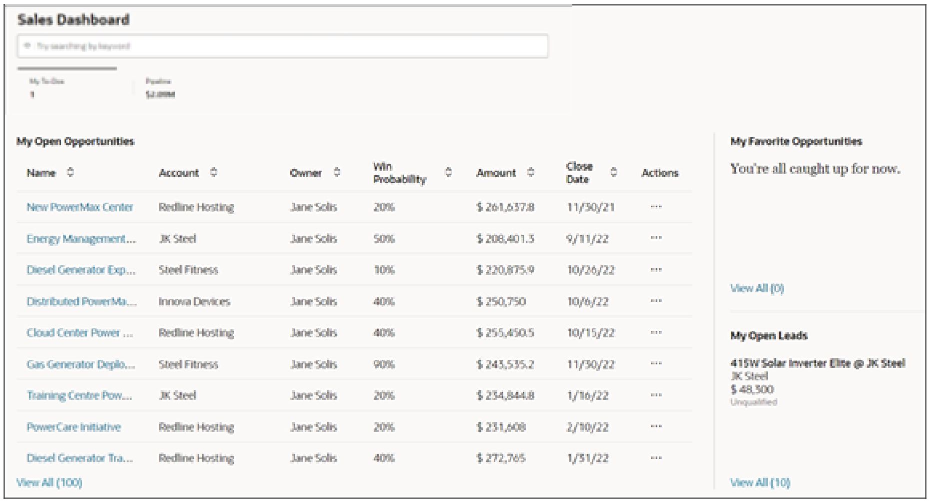
E-mail updated contract before the scheduled ...
Jane Solis
4/26/23
E-mail

Create quote for Pinnacle Technologies
Jane Solis
5/12/23
Text

The My To-Dos tab includes these components:

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

- **Pipeline**



Sales Dashboard

Try searching by keyword

My To-Dos 1

Pipeline $2.09M

My Open Opportunities

Name	Account	Owner	Win Probability	Amount	Close Date	Actions
New PowerMax Center	Redline Hosting	Jane Solis	20%	$ 261,637.8	11/30/21	...
Energy Management...	JK Steel	Jane Solis	50%	$ 208,401.3	9/11/22	...
Diesel Generator Exp...	Steel Fitness	Jane Solis	10%	$ 220,875.9	10/26/22	...
Distributed PowerMa...	Innova Devices	Jane Solis	40%	$ 250,750	10/6/22	...
Cloud Center Power ...	Redline Hosting	Jane Solis	40%	$ 255,450.5	10/15/22	...
Gas Generator Deplo...	Steel Fitness	Jane Solis	90%	$ 243,535.2	11/30/22	...
Training Centre Pow...	JK Steel	Jane Solis	20%	$ 234,844.8	1/16/22	...
PowerCare Initiative	Redline Hosting	Jane Solis	20%	$ 231,608	2/10/22	...
Diesel Generator Tra...	Redline Hosting	Jane Solis	40%	$ 272,765	1/31/22	...

View All (100)

My Favorite Opportunities

You're all caught up for now.

View All (0)

My Open Leads

415W Solar Inverter Elite @ JK Steel
JK Steel
$ 48,300
Unqualified

View All (10)

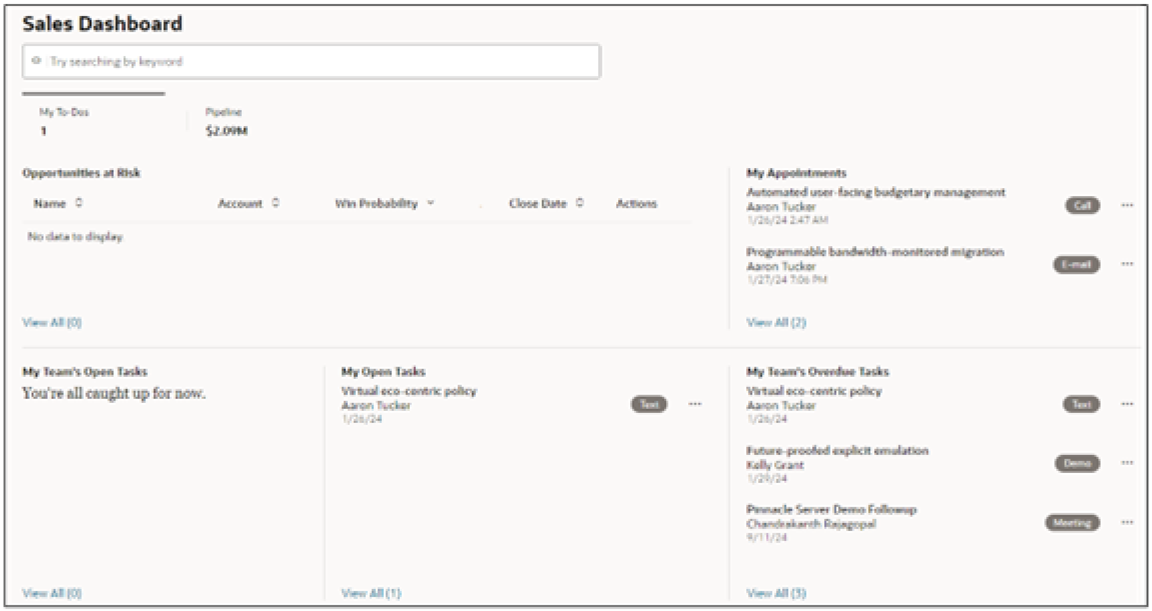
The Pipeline tab includes these components:

- 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-Do**



My To-Dos
1

Pipeline
$2.09M

Opportunities at Risk

Name	Account	Win Probability	Close Date	Actions
No data to display				

View All (0)

My Appointments

Automated user-facing budgetary management
Aaron Tucker
1/26/24 2:47 AM
Call

Programmable bandwidth-monitored migration
Aaron Tucker
1/27/24 7:06 PM
E-mail

View All (2)

My Team's Open Tasks

You're all caught up for now.

View All (0)

My Open Tasks

Virtual eco-centric policy
Aaron Tucker
1/26/24
Text

View All (1)

My Team's Overdue Tasks

Virtual eco-centric policy
Aaron Tucker
1/26/24
Text

Future-proofed explicit emulation
Kelly Grant
1/29/24
Demo

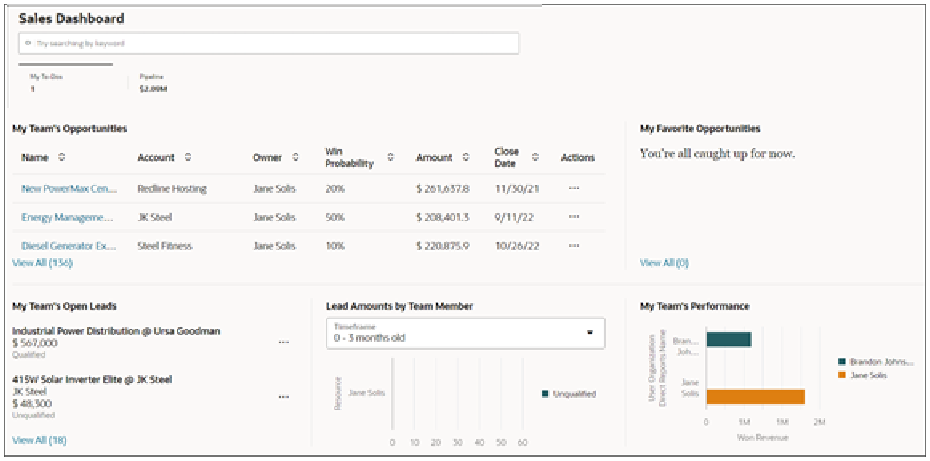
Pinnacle Server Demo Followup
Chandrakanth Rajagopal
9/11/24
Meeting

View All (3)

The sales manager To-Do 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**



Sales Dashboard

Try searching by keyword

My To-Dos: 1
Pipeline: $2.09M

My Team's Opportunities

Name	Account	Owner	Win Probability	Amount	Close Date	Actions
New PowerMax Cen...	Redline Hosting	Jane Solis	20%	$ 261,637.8	11/30/21	···
Energy Manageme...	JK Steel	Jane Solis	50%	$ 208,401.3	9/11/22	···
Diesel Generator Ex...	Steel Fitness	Jane Solis	10%	$ 220,875.9	10/26/22	···

View All (136)

My Favorite Opportunities

You're all caught up for now.

View All (0)

My Team's Open Leads

Industrial Power Distribution @ Ursa Goodman
$ 567,000
Qualified

415W Solar Inverter Elite @ JK Steel
JK Steel
$ 48,300
Unqualified

View All (18)

Lead Amounts by Team Member

Timeframe: 0 - 3 months old

My Team's Performance

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).

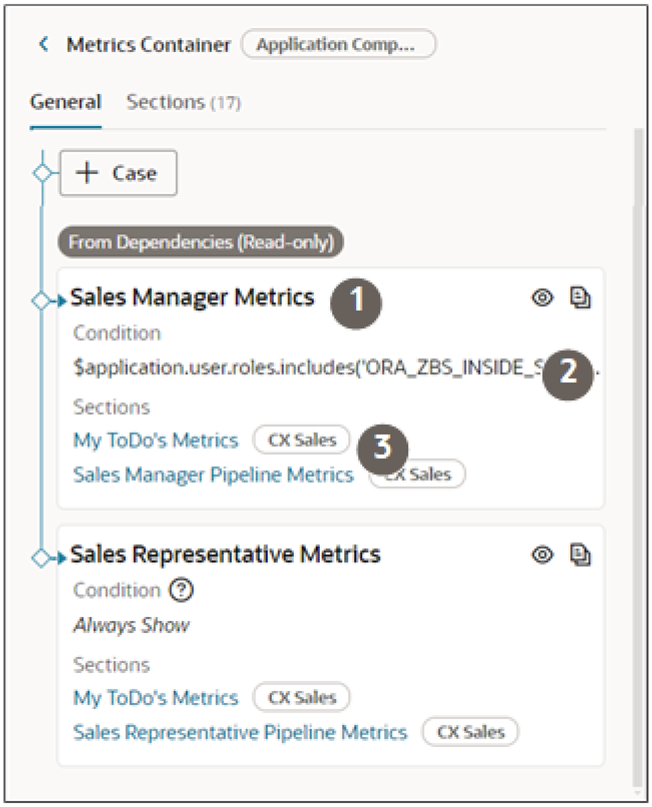
Each metrics container instance is composed of a list of Metric Cards under the Sections heading. Each metric card represents a tab on the dashboard. The order of the sections determines the order of the tabs.

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.



Metrics Container (Application Comp...)

General | Sections (17)

+ Case

From Dependencies (Read-only)

Sales Manager Metrics

Condition

$application.user.roles.includes('ORA_ZBS_INSIDE_S...

Sections

My ToDo's Metrics (CX Sales)

Sales Manager Pipeline Metrics (CX Sales)

Sales Representative Metrics

Condition

Always Show

Sections

My ToDo's Metrics (CX Sales)

Sales Representative Pipeline Metrics (CX Sales)

Metric Cards and Contents Containers

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

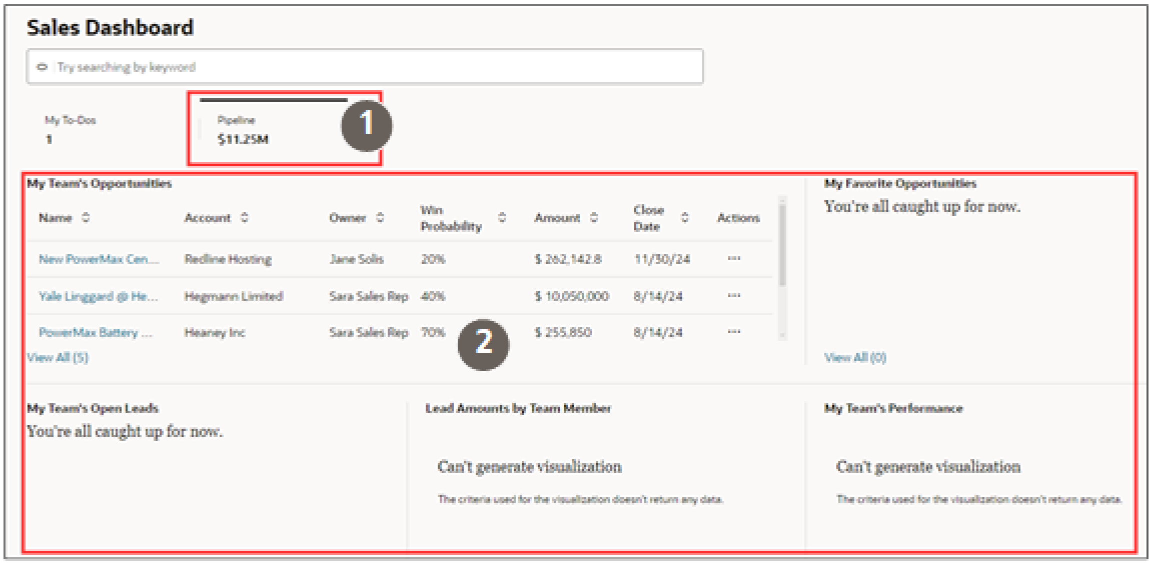
- **Metrics Card** (identified by callout 1 in the following screenshot)

  A metric card holds the information presented at the top of the tab, which is always visible on the dashboard even when the tab isn't selected. It contains the title and an optional rollup of key information presented in the tab. The predefined Pipeline metric container for sales managers, for example, calculates and displays the sum of opportunity revenue from the My Team's Opportunities saved search. Optional badges can alert a salesperson if the information needs attention.

- **Contents Container** (callout 2)

  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. The table information is drawn from the same My Teams Opportunities saved search as the metrics container. 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.



Sales Dashboard

Try searching by keyword

My To-Dos
1

Pipeline
$11.25M

My Team's Opportunities

Name	Account	Owner	Win Probability	Amount	Close Date	Actions
New PowerMax Cen...	Redline Hosting	Jane Solis	20%	$ 262,142.8	11/30/24	···
Yale Linggard @ He...	Hegmann Limited	Sara Sales Rep	40%	$ 10,050,000	8/14/24	···
PowerMax Battery ...	Heaney Inc	Sara Sales Rep	70%	$ 255,850	8/14/24	···

View All (5)

My Favorite Opportunities

You're all caught up for now.

View All (0)

My Team's Open Leads

You're all caught up for now.

Lead Amounts by Team Member

Can't generate visualization

The criteria used for the visualization doesn't return any data.

My Team's Performance

Can't generate visualization

The criteria used for the visualization doesn't return any data.

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

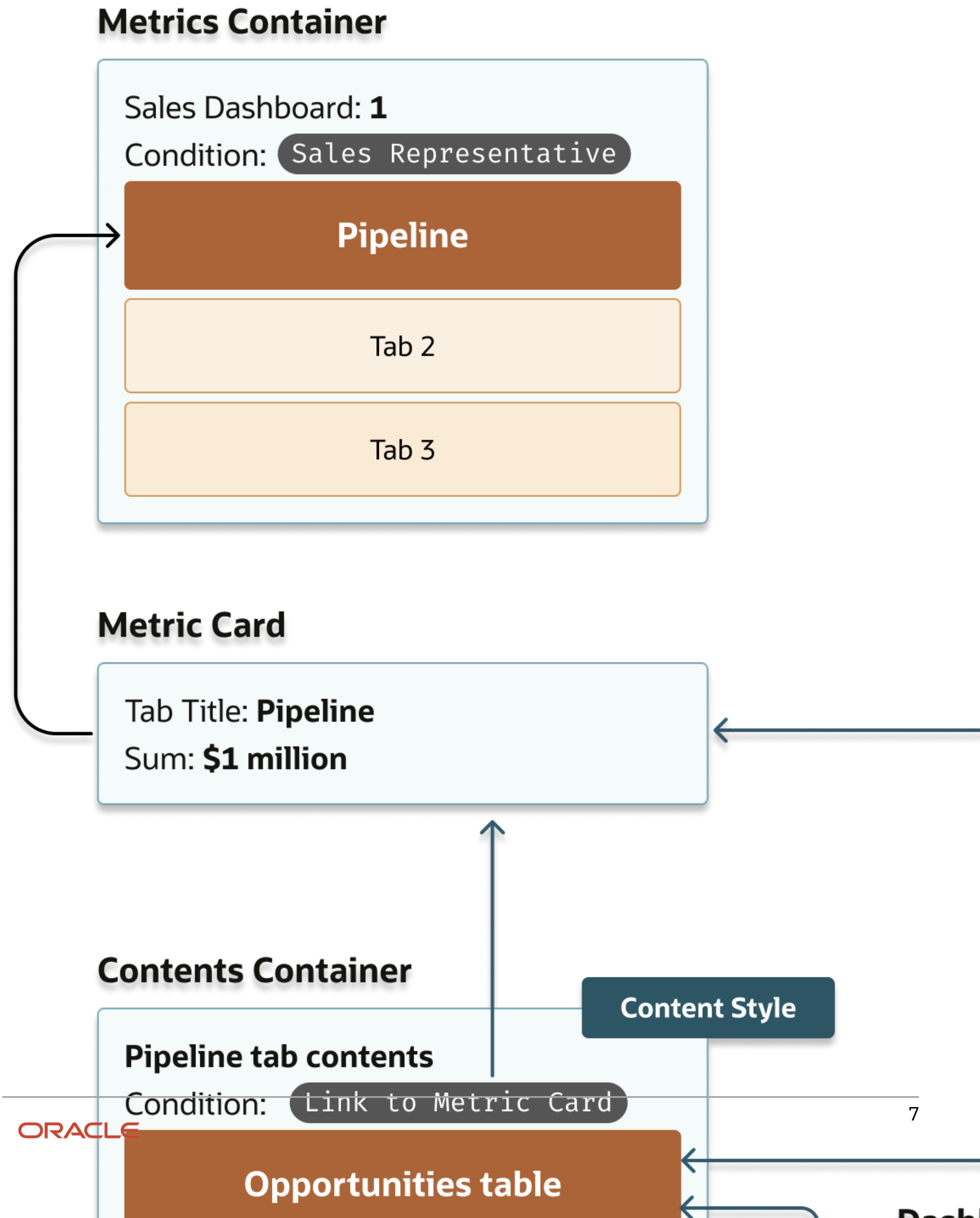
Here's an overview of the main VB Studio components that you use to create a custom dashboard. This example focuses on creating a tab displaying a table of opportunity pipeline information from a saved search.

- 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**.

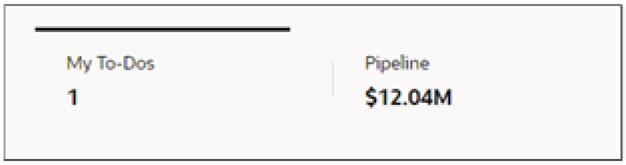

Metrics Container
Sales Dashboard: 1
Condition: Sales Representative
Pipeline
Tab 2
Tab 3
Metric Card
Tab Title: Pipeline
Sum: $1 million
Contents Container
Content Style
Pipeline tab contents
Condition: Link to Metric Card
Opportunities table

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 saved searches or OTBI analyses, but the calculation is done in the code you add.



My To-Dos
1

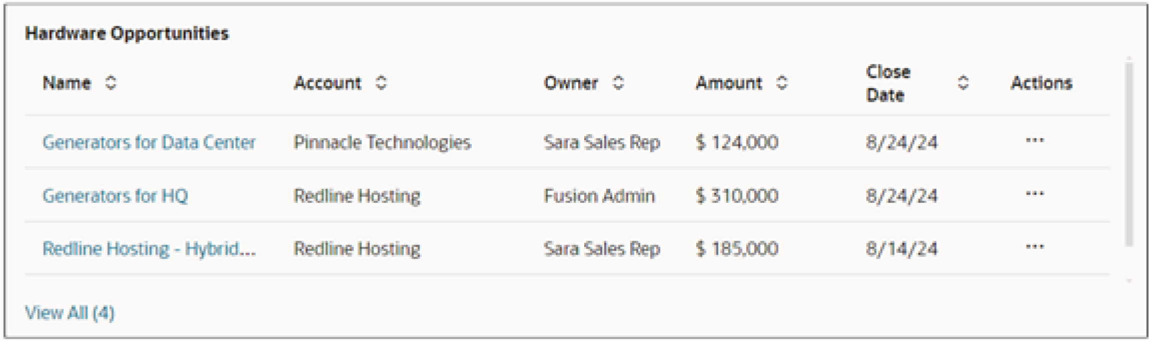
Pipeline
$12.04M

Tables

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

The tables you display from saved searches make it possible for salespeople to drill down into the individual records in the table and take actions.

To display an OTBI analysis as a table, you must first create a Tabular Chart visualization using the Visualization Configuration tool and then display that visualization. Tabular Charts don't permit drill-downs.



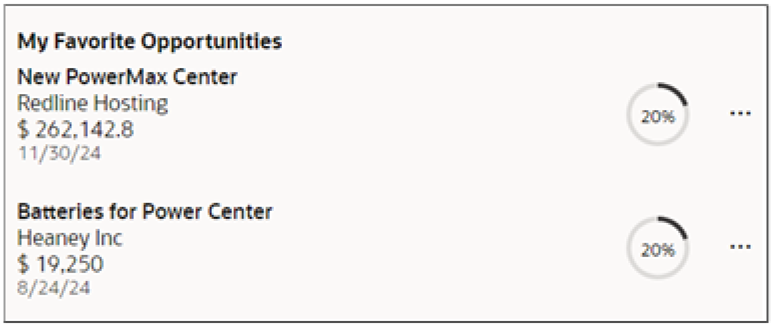
Hardware Opportunities

Name	Account	Owner	Amount	Close Date	Actions
Generators for Data Center	Pinnacle Technologies	Sara Sales Rep	$ 124,000	8/24/24	···
Generators for HQ	Redline Hosting	Fusion Admin	$ 310,000	8/24/24	···
Redline Hosting - Hybrid...	Redline Hosting	Sara Sales Rep	$ 185,000	8/14/24	···

View All (4)

Lists

You can display saved searches as a list.



My Favorite Opportunities

New PowerMax Center
Redline Hosting
$ 262,142.8
11/30/24

20%

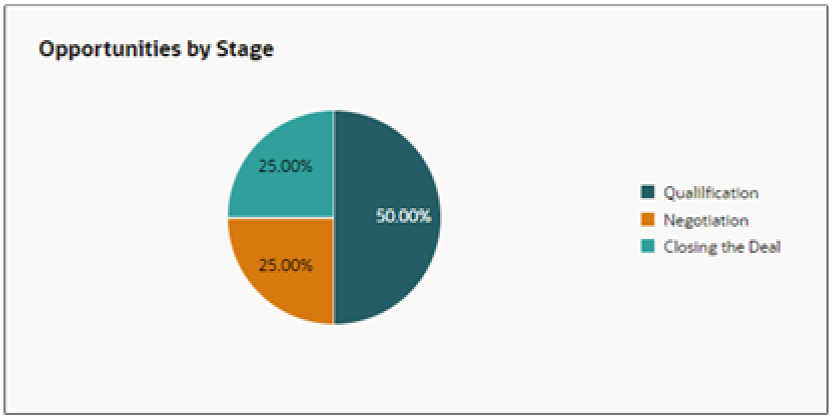
Batteries for Power Center
Heaney Inc
$ 19,250
8/24/24

20%

Charts

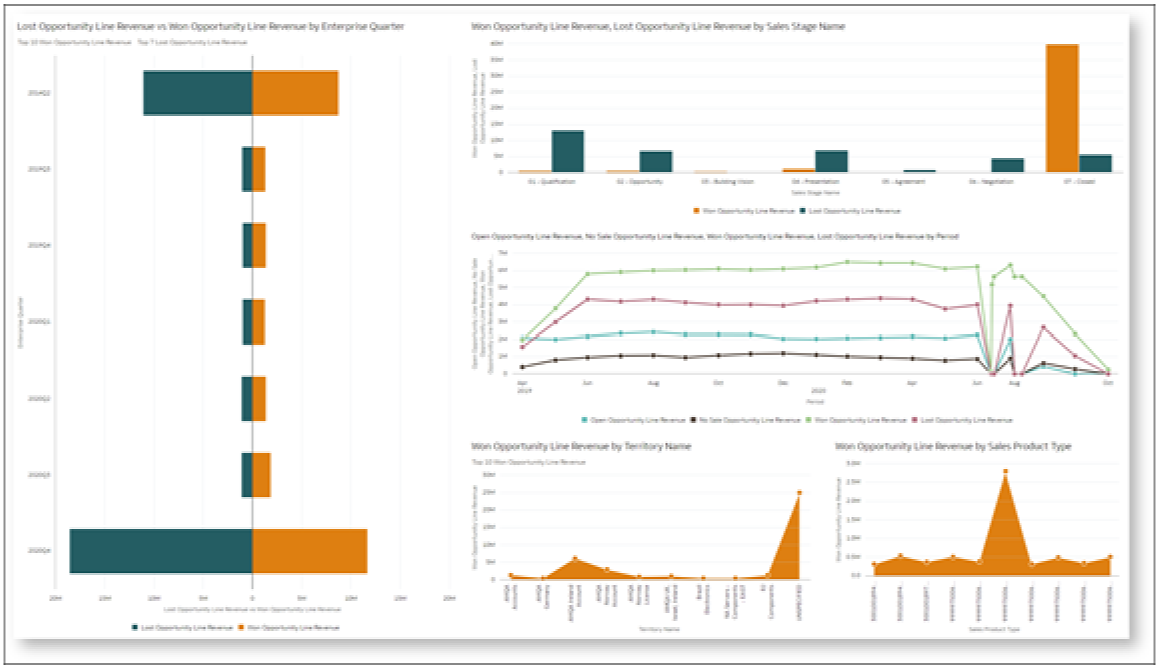
You can display charts that you created using the Visualization Configuration tool and charts from Express Reports. You can add charts from Express Report directly. To include charts from saved searches and OTBI analyses, you create visualizations using the Visualization Configuration tool first.

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.


Lost Opportunity Line Revenue vs Won Opportunity Line Revenue by Enterprise Quarter
Won Opportunity Line Revenue, Lost Opportunity Line Revenue by Sales Stage Name
01 - Qualification
02 - Opportunity
03 - Building Vision
04 - Presentation
05 - Agreement
06 - Negotiation
07 - Closed
Sales Stage Name
Won Opportunity Line Revenue
Lost Opportunity Line Revenue
Open Opportunity Line Revenue, No Sale Opportunity Line Revenue, Won Opportunity Line Revenue, Lost Opportunity Line Revenue by Period
Won Opportunity Line Revenue by Territory Name
Won Opportunity Line Revenue by Sales Product Type

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 summaries the data in saved searches to the metric card at the top of each tab.
  - You can display saved searches as a table or a list.

    > **Note:** Only the filters are carried over from the saved search to the sales dashboard table. You must use VB Studio to specify which columns you want to display and their sort order in the dashboard.

  - 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 summaries of sales data 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.

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

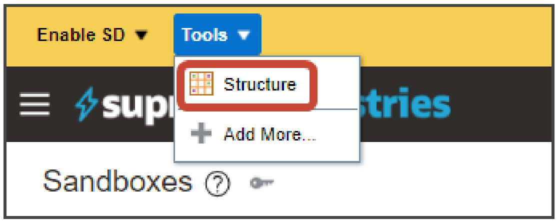
At some point in your extension lifecycle, you might need to create a new workspace in an existing project. You may want to create a new workspace from the main branch if you forget what changes a particular workspace contains, for example. Follow the instructions in the topic: *Clone an Existing Repository*.

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

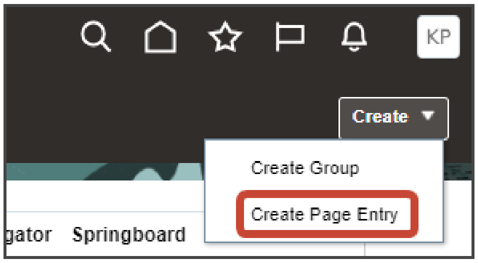
Enable Navigation to the Sales Dashboard

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:

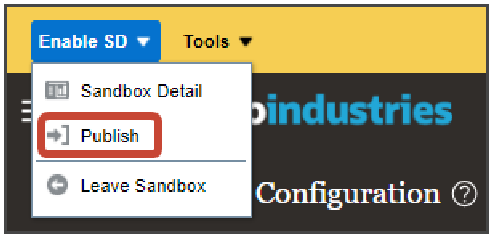
Create Page Entry Field Values for Sales Dashboard

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

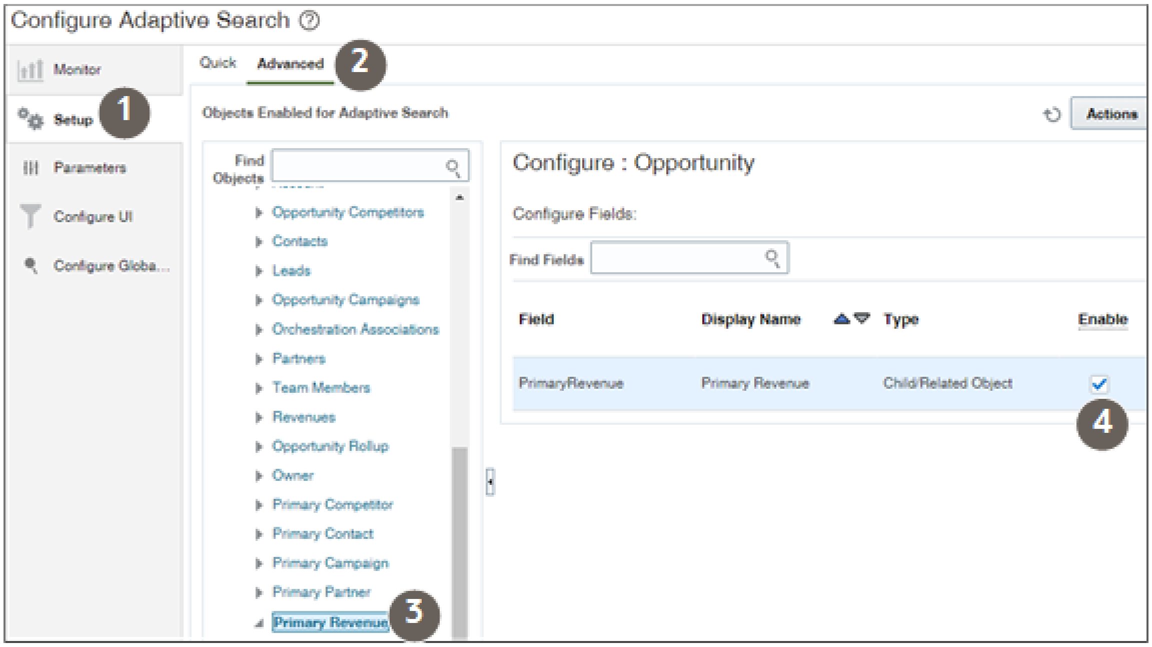
  Navigation: **Opportunity** > **Primary Revenue** > **Primary Revenue**
- Win Probability

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

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

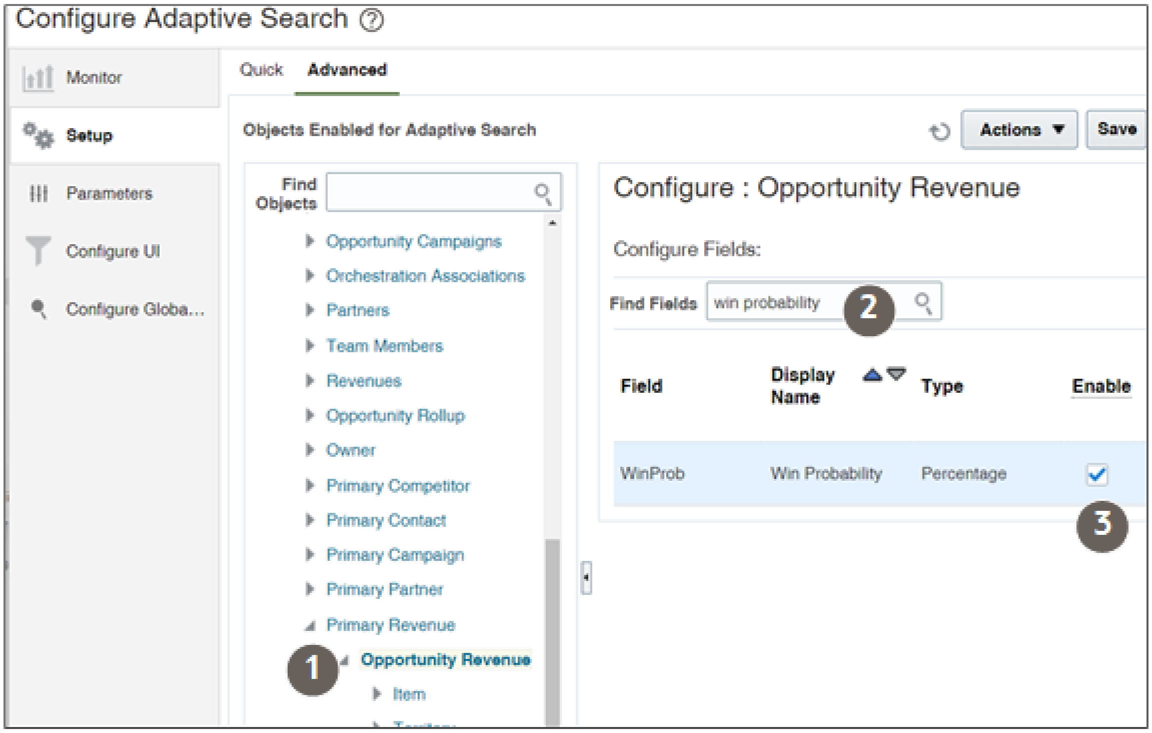
Here's how to enable the fields:

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.



Field	Display Name	Type	Enable
PrimaryRevenue	Primary Revenue	Child/Related Object	✓

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**.

3 Visualization Configurations

Create Visualizations

Using the Visualization Configurations feature in Application Composer, you can create a visualization from a Saved Search or from an Oracle Transactional Business Intelligence (OTBI) analysis. A visualization is a visual representation of your data, such as a bar chart, pie chart, or a funnel chart. You can add visualization to the tabs in the Sales Dashboard or to the CX Sales Mobile app.

You can also create a visualization to display an OTBI analysis as a table (Tabular Chart).

> **Note:** Tabular charts aren't available for saved searches. That's because you can display saved searches as a table directly as described in the topic *Display a Saved Search as a Table*. Displaying saved searches as tables directly enables drill-downs. The tables created as visualizations don't.

Prerequisites

Users who create visualizations, must have the privilege ZCA_VIEW_DATA_VISUALIZATION_CONFIGURATION_PRIV.

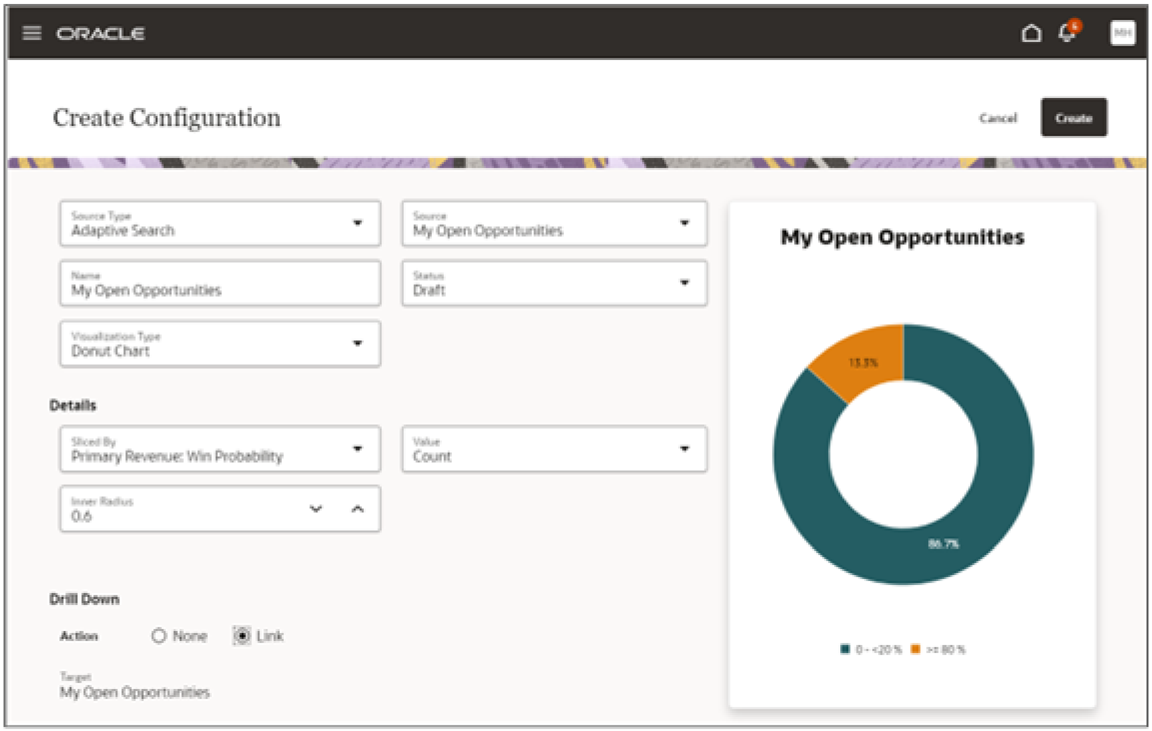
Setup users with one of the following job roles provided by Oracle already have this privilege:

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

Create a Visualization from a 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. From the Create Configuration page, **Source Type** field, select the **Adaptive Search**.



Create Configuration

Source Type: Adaptive Search
Source: My Open Opportunities
Name: My Open Opportunities
Status: Draft
Visualization Type: Donut Chart

Details

Sliced By: Primary Revenue: Win Probability
Value: Count
Inner Radius: 0.6

Drill Down

Action: ○ None ● Link

Target: My Open Opportunities

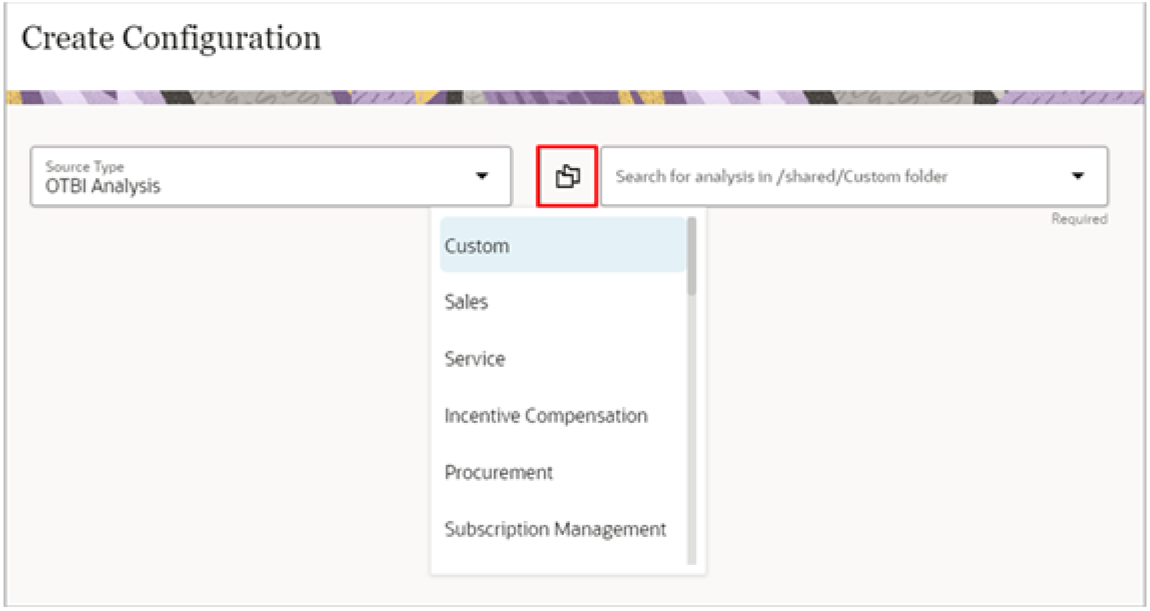
5. In the Visualization Configurations page, click **Add**.
6. In the **Source Type** field, select **Adaptive Search**.
7. In the **Source** field, select the saved search. The saved search is used as the data source for the visualization.
8. In the **Visualization Type** field, choose your preferred 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.
10. Enable salespeople to drill down to the list page with a list of items in each section of the chart. Select **Drill Down** > **Link**.
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.

The chart is now ready to be added to the Sales Dashboard, or to the CX Sales Mobile dashboard. For mobile setup see the topic *Configure the Reports Page Layouts*

You can display the saved search visualization on the CX Sales Mobile application Home page as a card. See the topic: *Display Visualizations from Saved Searches in Home Page Cards and the Reports List*.

How to Create a Visualization from an OTBI Analysis

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 a source analysis from the search results. All the fields get selected automatically, and the visualization shows on the page. If the selected source analysis has more columns than required, there could be a warning message that the visualization might not show the correct data. To resolve this issue, remove the columns that aren't used, and reload the configuration.
9. In the **Name** field, enter the display title of the chart.
10. In the **Visualization Type** field, select the chart type. To display the OTBI analysis as a table, select **Tabular Chart**.
11. Different types of visualization require you to enter different parameters. For example:
    - For a Tabular Chart, you can select alternate columns to display in the table and change their order.
    - For a Combo Chart, you can have a stacked bar chart and you can have two Y axes with different metrics.
12. To enable drill-down functionality:
    a. Turn on **Drill Down**.
    b. Click the **Select Catalog Folder** icon and select the folder where you've saved the target OTBI analysis. You can drill down to the same or different analysis. For example, you can display a more detailed analysis when salespeople click a chart section.
    c. In the **Target** field, search for and select the target OTBI analysis . The target analysis must have a prompted filter applied on the same field as selected in the source analysis. The target analysis opens the target analysis in a new window.

       To create a configuration using a predefined BI Analysis, select **Sales** as the Catalog folder and then search for the source.
    d. After enabling drill-down, you can validate the drill-down functionality by clicking on any component of the visualization.
13. Click **Create** to save the configuration.
14. 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 I group data in saved searches and visualizations by team members?

To group data, including by the different team members in saved searches and their visualizations, you 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.

Here's the setup overview:

1. If you're using Workspace in the classic Sales UI, then enable the Group By feature for Workspace. This step isn't required for Sales in the Redwood User Experience.

   See the *Enable the Group By Feature for Workspace* section for detailed steps.
2. Enable the Team Members and Team Member fields for Group By in the Configure UI tab on the Configure Adaptive Search page.

   See the *Enable Team Member Fields for Group By* section for detailed steps.
3. Create the saved search with the Team Members group by selected.

   See the topic *Create Saved Searches for the Sales Organization*.
4. Create the visualization from the saved search as described in the topic *Create Visualizations*.

Enable the Group By Feature for Workspace

1. Click **Navigator** > **My Enterprise** > **Setup and Maintenance**.
2. In the Setup and Maintenance work area, open the **Configure Adaptive Search** task.

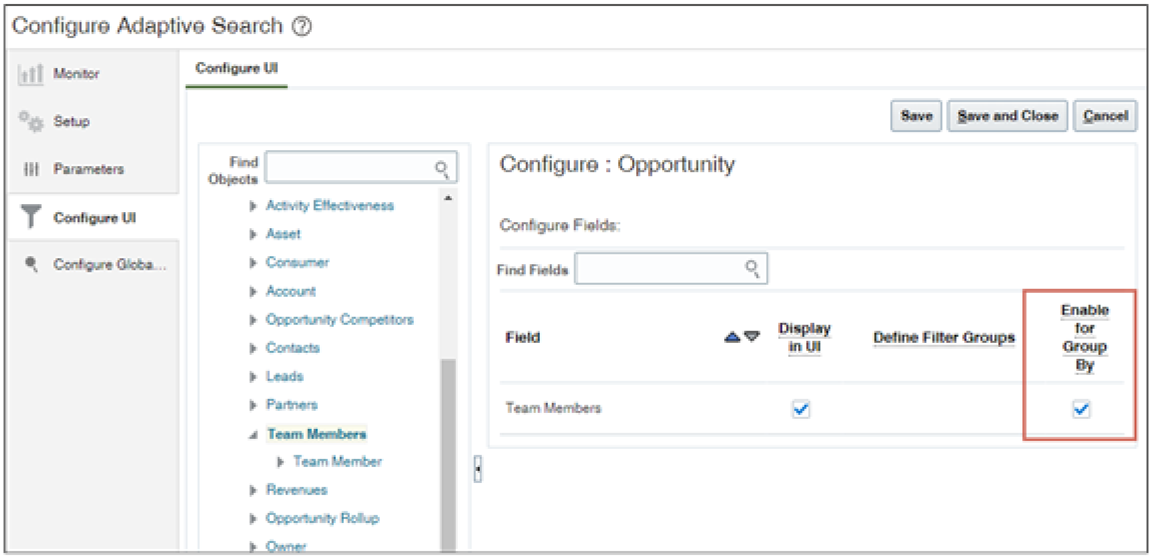
   - Offering: Sales
   - Functional Area: Sales Foundation

- 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.

No publishing is required to see the option in the group by. The changes you made are reflected in the Workspace UI for the Group By field drop down lists for the selected object.

Related Topics

- Create Visualizations
- Create Saved Searches for the Sales Organization

4 Configure the Dashboard

Overview of Sales Dashboard Configuration Steps

Here's an overview of the steps to create 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	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 One of the Dashboards Provided by Oracle*
2	Within the dashboard shell container, create the metric card for the tab you're adding. The metric card is the top of the tab that's always visible.	*Create a Metric Card for the Dashboard Tab*
3	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 Metric Card*
4	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 its position.	*How Information Displays in a Sales Dashboard Tab as You Add Sections*
5	Adding sections predefined by Oracle is as easy as clicking **Add** and selecting the content.	*Add Predefined Content to Your Tab*
6	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*

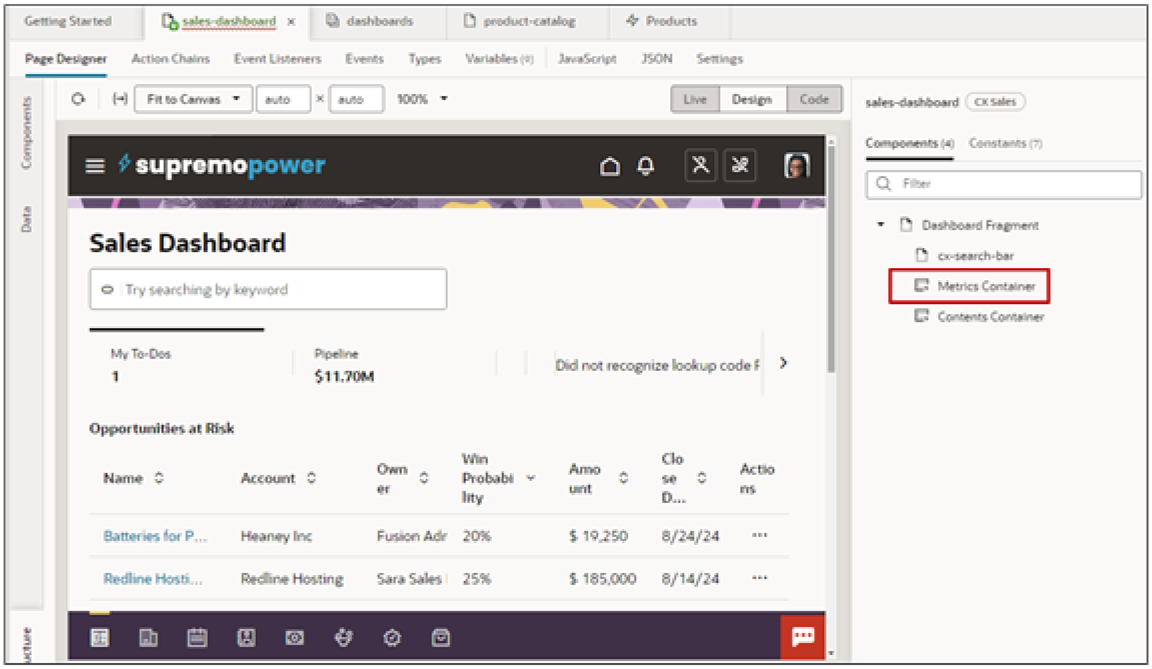
Step	Description	More Details
7	The parameters you enter are different for displaying a table, a list, a visualization, and for embedding a report from Oracle Fusion Data Intelligence. The values of some parameters differ by business object. The sample code provided in the topics is specific to opportunities. 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* - *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*
8	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*
9	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*

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

Before you can make any changes to the Sales Dashboard, you must make a copy of one of the dashboards provided by Oracle, the two instances of the Metrics Container. 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.



Sales Dashboard

Try searching by keyword

My To-Dos
1

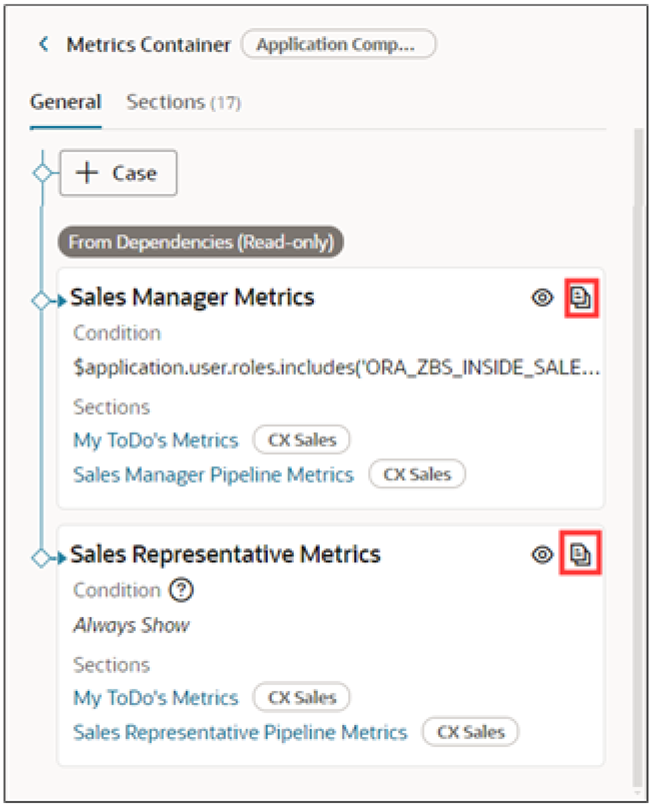
Pipeline
$11.70M

Did not recognize lookup code F

Opportunities at Risk

Name	Account	Owner	Win Probability	Amount	Close D...	Actions
Batteries for P...	Heaney Inc	Fusion Adr	20%	$ 19,250	8/24/24	···
Redline Hosti...	Redline Hosting	Sara Sales	25%	$ 185,000	8/14/24	···

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



Sales Manager Metrics

Condition

$application.user.roles.includes('ORA_ZBS_INSIDE_SALE...

Sections

My ToDo's Metrics CX Sales

Sales Manager Pipeline Metrics CX Sales

Sales Representative Metrics

Condition

Always Show

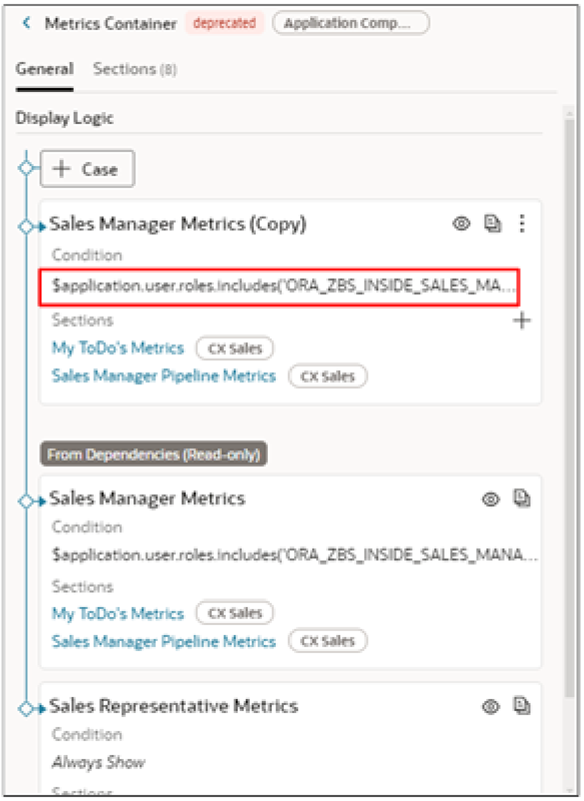
Sections

My ToDo's Metrics CX Sales

Sales Representative Pipeline Metrics CX Sales

4. To enable you to view your sales dashboard during setup, edit the **Condition** field or 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.



Sales Manager Metrics (Copy)
Condition
$application.user.roles.includes('ORA_ZBS_INSIDE_SALES_MA...
Sections
My ToDo's Metrics CX Sales
Sales Manager Pipeline Metrics CX Sales

From Dependencies (Read-only)

Sales Manager Metrics
Condition
$application.user.roles.includes('ORA_ZBS_INSIDE_SALES_MANA...
Sections
My ToDo's Metrics CX Sales
Sales Manager Pipeline Metrics CX Sales

Sales Representative Metrics
Condition
Always Show
Sections

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 right position in the list of cases.

   For example, the Sales Manager Metrics dashboard provided by Oracle, displays only 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 create your own metric cards as described in the topic *Create a Metric Card for the Dashboard Tab*

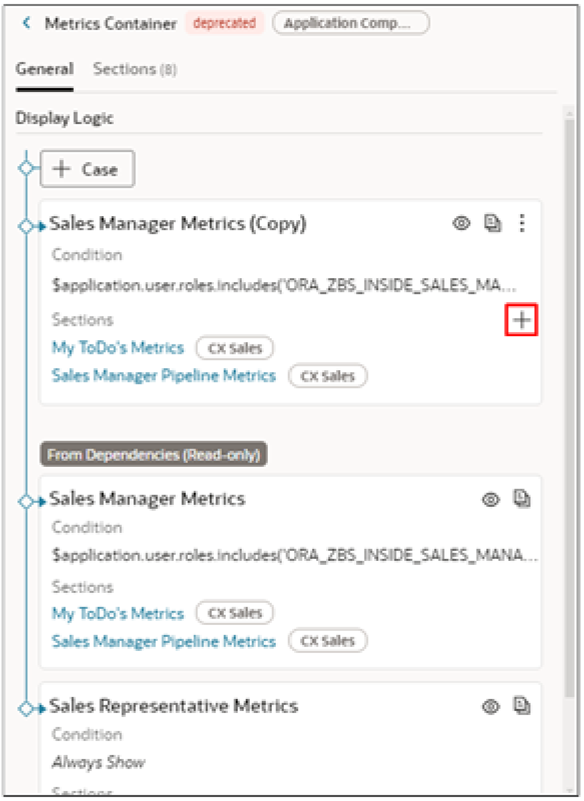
   You can also edit the existing dashboard contents. You can:

   - Delete any of the existing metric cards.
   - Change their position on the tab using the arrow icons.

Create a Metric Card for the Dashboard Tab

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

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



Sales Manager Metrics (Copy)
Condition
$application.user.roles.includes('ORA_ZBS_INSIDE_SALES_MA...
Sections
My ToDo's Metrics
Sales Manager Pipeline Metrics

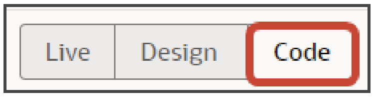
Sales Manager Metrics
Condition
$application.user.roles.includes('ORA_ZBS_INSIDE_SALES_MANA...
Sections
My ToDo's Metrics
Sales Manager Pipeline Metrics

Sales Representative Metrics
Condition
Always Show

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.

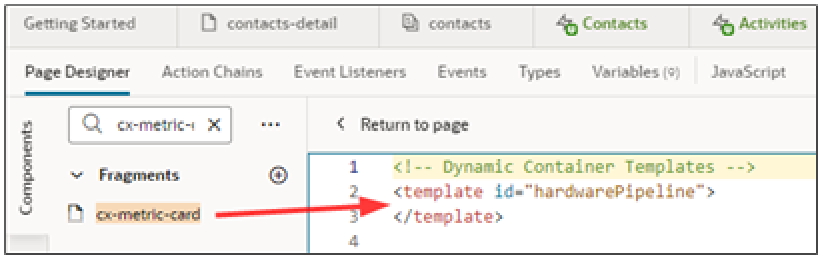
3. In the Create Section window, enter a title that's visible only during setup. For example, enter **Sales Manager Hardware Pipeline**. The title you enter here isn't the title that appears to salespeople at runtime. You'll enter the actual tab title as a parameter in the code.
4. Drill down into the new section you created.
5. Click **Code**.



6. Your new **Sales Manager Hardware Pipeline** section displays with 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.

   ```
   <!-- Dynamic Container Templates -->
   <template id="hardwarePipeline">
   </template>
   ```

   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 topic: *Add Badges to Highlight Important Dashboard Information*.

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**.

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*.

Add Badges to Highlight Important Dashboard Information

Here's how to configure badges for your 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:

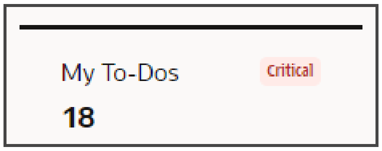
- If fewer than 5 overdue tasks exist, then don't show a badge at all.
- If 5-10 overdue tasks exist, then show a warning badge.
- If more than 10 overdue tasks exist, then show a critical badge.

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 • info • neutral • success • warning Each type of badge (`status`) displays with 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**.

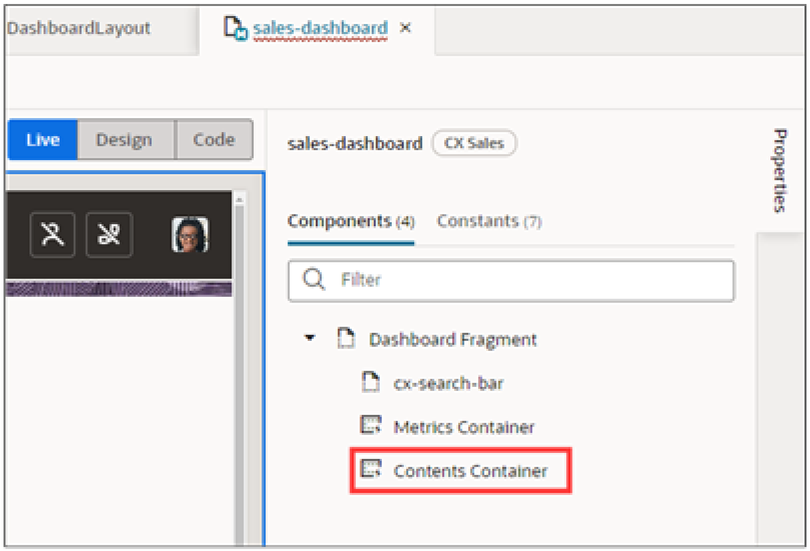
Tip:

- 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

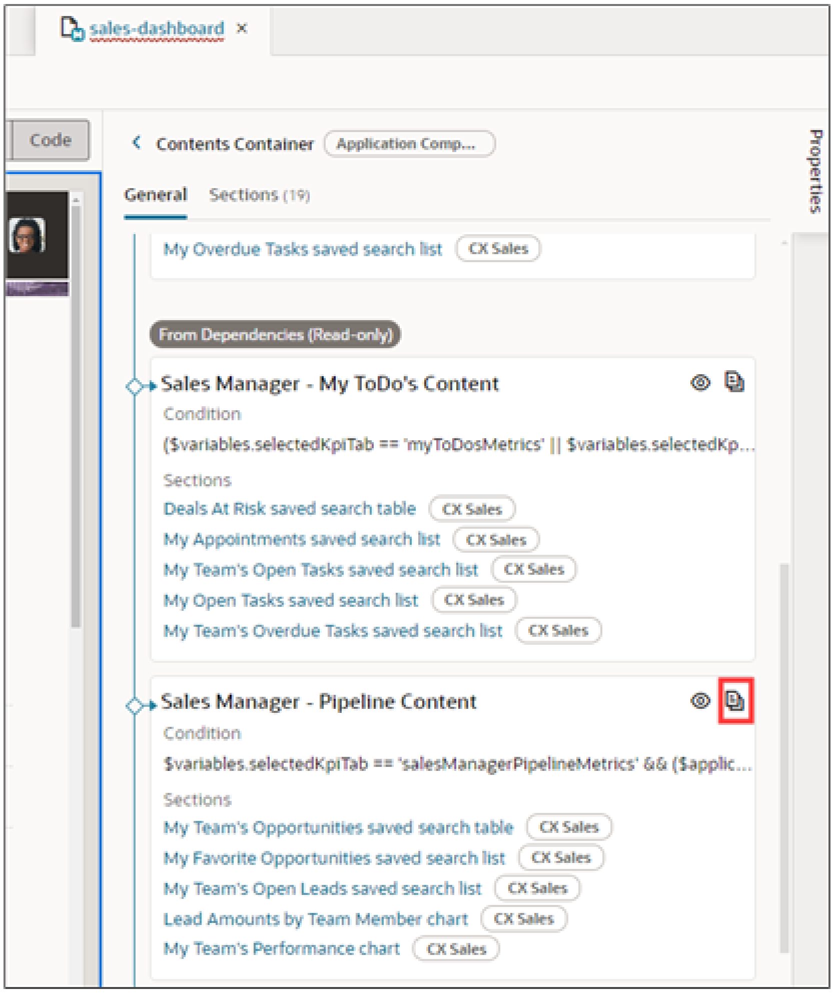
After you've created a custom metric card for your dashboard tab, you're ready to create the container that will hold the content you want to display. Here's how.

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**.



My Overdue Tasks saved search list CX Sales

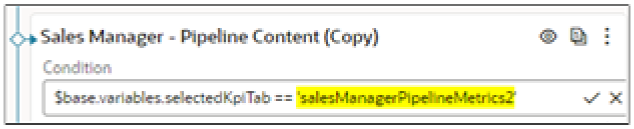
Sales Manager - My ToDo's Content
Condition
($variables.selectedKpiTab == 'myToDosMetrics' || $variables.selectedKp...
Sections
Deals At Risk saved search table CX Sales
My Appointments saved search list CX Sales
My Team's Open Tasks saved search list CX Sales
My Open Tasks saved search list CX Sales
My Team's Overdue Tasks saved search list CX Sales

Sales Manager - Pipeline Content
Condition
$variables.selectedKpiTab == 'salesManagerPipelineMetrics' && ($applic...
Sections
My Team's Opportunities saved search table CX Sales
My Favorite Opportunities saved search list CX Sales
My Team's Open Leads saved search list CX Sales
Lead Amounts by Team Member chart CX Sales
My Team's Performance chart CX Sales

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.



Sales Manager - Pipeline Content (Copy)
Condition
$base.variables.selectedKpiTab == 'salesManagerPipelineMetrics2'

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.

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

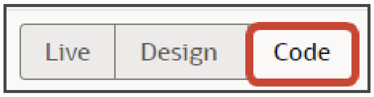
1. If the copy of the content container, already includes 5 sections, you must delete a section to add a new one. To delete a section, select a row and click **Remove Item** (the trash can icon).
2. In the custom content container, click **Add Section**.
3. To add any of the content predefined by Oracle 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**.

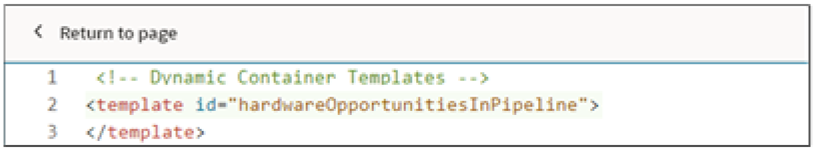
    b. In the Create Section window, enter a title for the section, for example: **Hardware Opportunities Pipeline**. The title is visible only during setup and as the ID in the code.

    c. Drill down into the copy.

    d. Click **Code**.

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

    ```
    <!-- Dynamic Container Templates -->
    <template id="hardwareOpportunitiesInPipeline">
    </template>
    ```

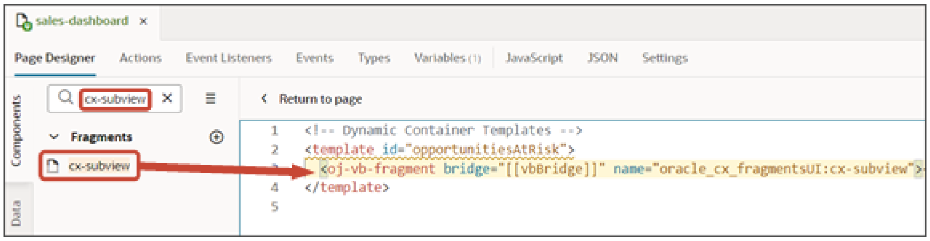
    e. Enter the appropriate code between the template tags. The code you enter is different for tables, lists, visualizations and for embedded Oracle Fusion Data Intelligence content. For details see these topics:

    - *Display a Saved Search as a Table*
    - *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*

Display a Saved Search as a Table

If your data comes from a saved search, you can display the data as a table in your sales dashboard.

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.



```
<!-- Dynamic Container Templates -->
<template id="opportunitiesAtRisk">
  <oj-vb-fragment bridge="[[vbBridge]]" name="oracle_cx_fragmentsUI:cx-subview">
</template>
```

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>
```

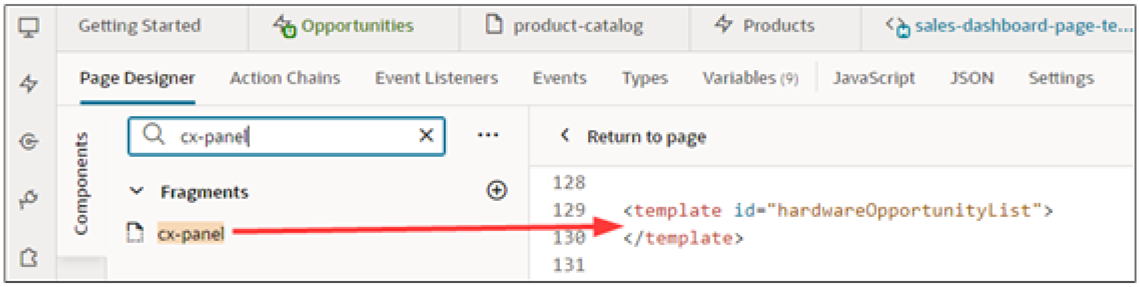
Here is sample code with the child fragment added:

```
<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.



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

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

A 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

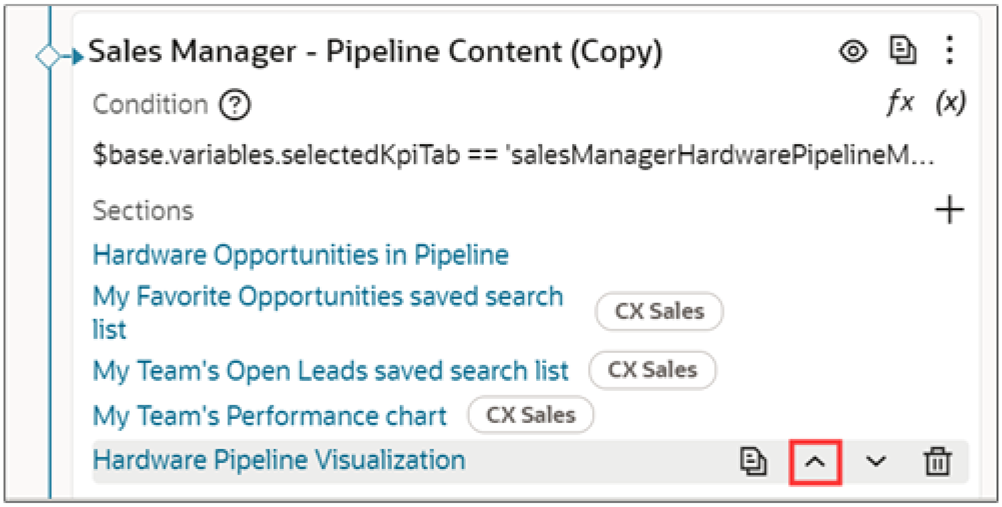
Here's how to add visualizations that you created using the Visualization Configurations page, or visualizations that are part of Express Reports.

> **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, you must delete one 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 on the new section 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.



Sales Manager - Pipeline Content (Copy)

Condition ⓘ

$base.variables.selectedKpiTab == 'salesManagerHardwarePipelineM...

Sections +

Hardware Opportunities in Pipeline

My Favorite Opportunities saved search list CX Sales

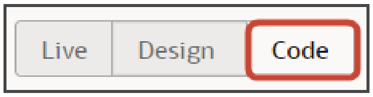
My Team's Open Leads saved search list CX Sales

My Team's Performance chart CX Sales

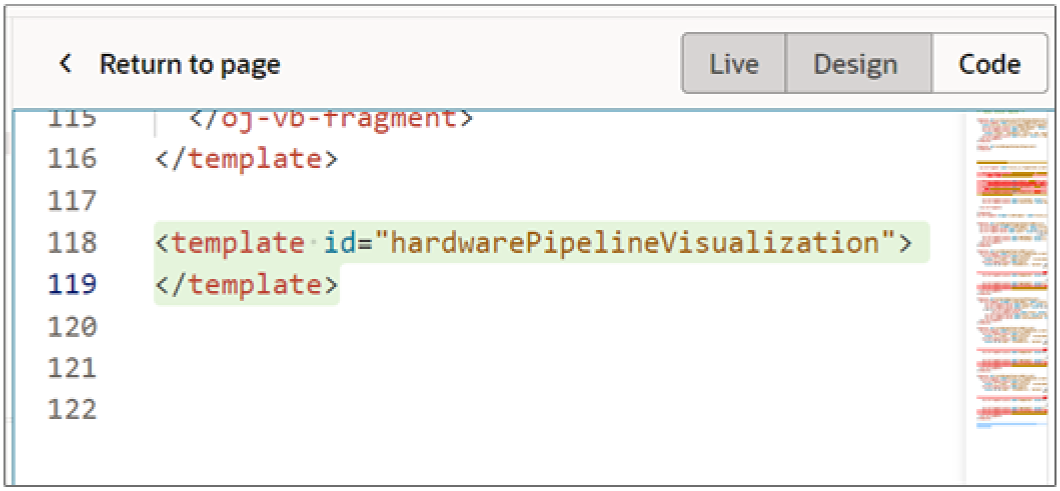
Hardware Pipeline Visualization

4. Click the section name link.

5. Click the **Code** button.



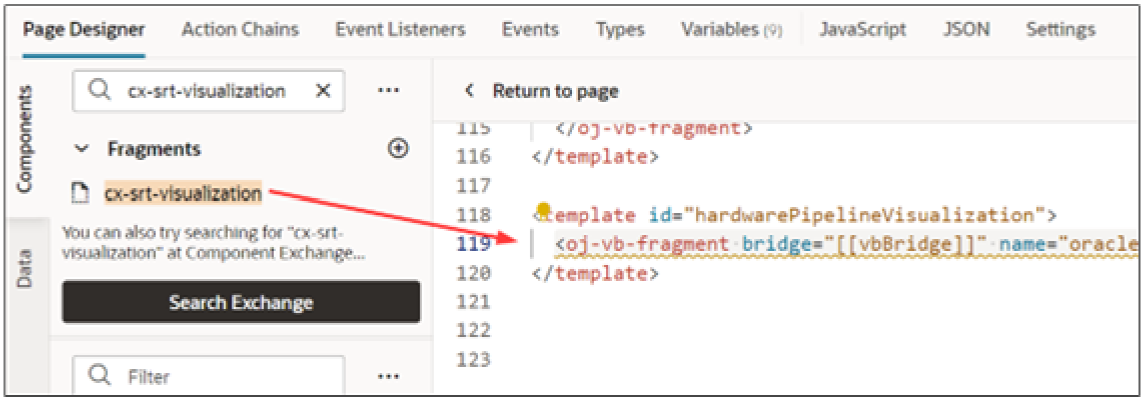
Your new section displays with empty placeholder `template` tags.



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

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

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.



You can also try searching for "cx-srt-visualization" at Component Exchange...

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

<template id="hardwarePipelineVisualization">
  <oj-vb-fragment bridge="[[vbBridge]]" name="oracle
</template>
```

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

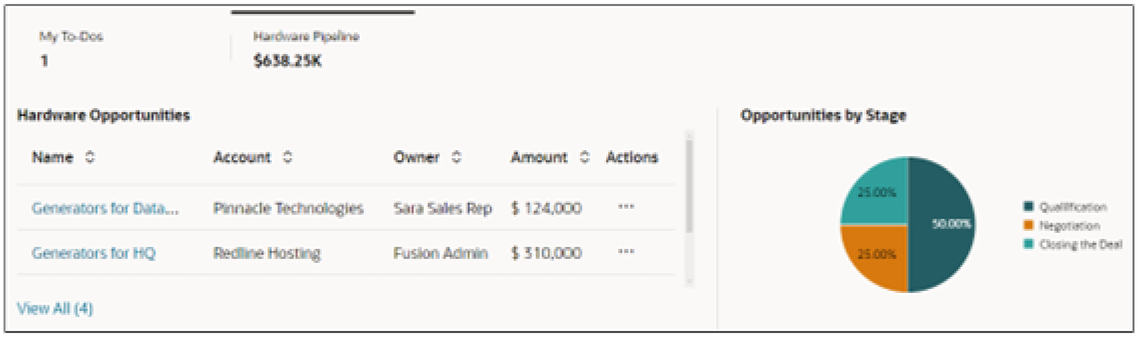
Parameter Name	Description
reportNumber	Enter the reference number of the visualization that you created using the Visualization Configuration tool. You can also display the visualizations created in Express Reports, by entering its report number. The sales dashboard only displays the visualization part 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.



My To-Dos
1

Hardware Pipeline
$638.25K

Hardware Opportunities

Name	Account	Owner	Amount	Actions
Generators for Data...	Pinnacle Technologies	Sara Sales Rep	$ 124,000	···
Generators for HQ	Redline Hosting	Fusion Admin	$ 310,000	···

View All (4)

Opportunities by Stage

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:

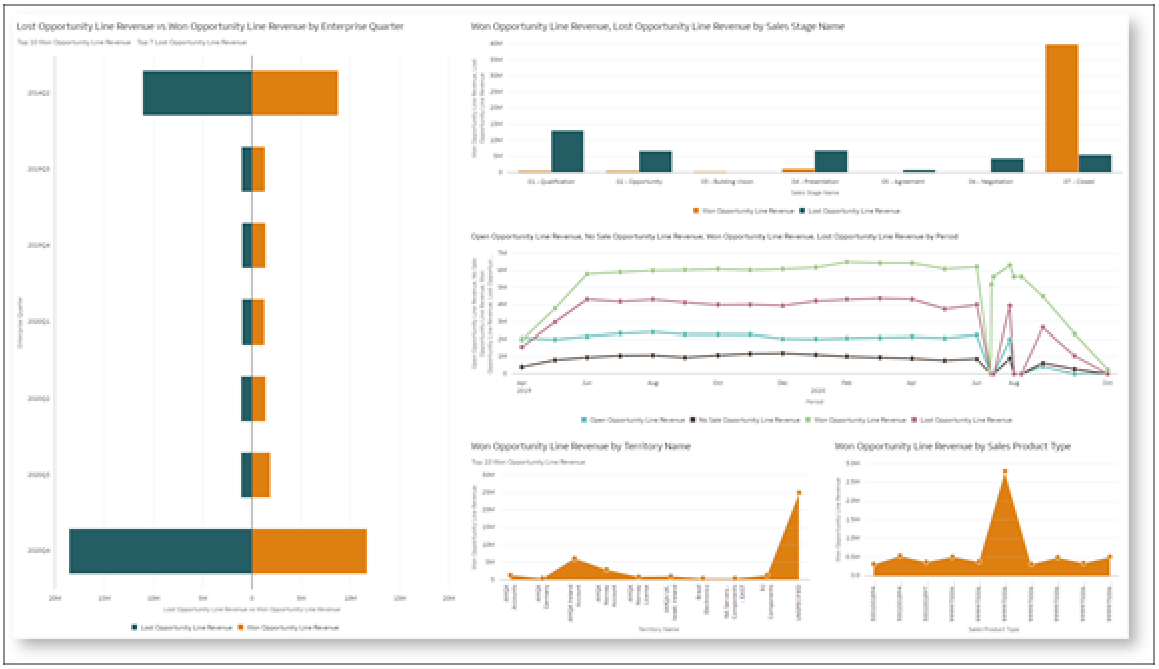
```
<template id="myList">
<oj-vb-fragment name="oracle_cx_fragmentsUI:cx-heterogeneous-list"
 class="oj-flex oracle-cx-fragmentsUI-cx-fragment-full-height" bridge="[[ vbBridge ]]">
 <oj-vb-fragment-param name="query" value="[[ [{type: 'qbe', params: [{key: '_userRelevantItems', value:
 true}]}] ]]">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


Lost Opportunity Line Revenue vs Won Opportunity Line Revenue by Enterprise Quarter
Won Opportunity Line Revenue, Lost Opportunity Line Revenue by Sales Stage Name
Won Opportunity Line Revenue
Lost Opportunity Line Revenue
Open Opportunity Line Revenue, No Sale Opportunity Line Revenue, Won Opportunity Line Revenue, Lost Opportunity Line Revenue by Period
Open Opportunity Line Revenue
No Sale Opportunity Line Revenue
Won Opportunity Line Revenue
Lost Opportunity Line Revenue
Won Opportunity Line Revenue by Territory Name
Won Opportunity Line Revenue by Sales Product Type
Lost Opportunity Line Revenue
Won Opportunity Line Revenue

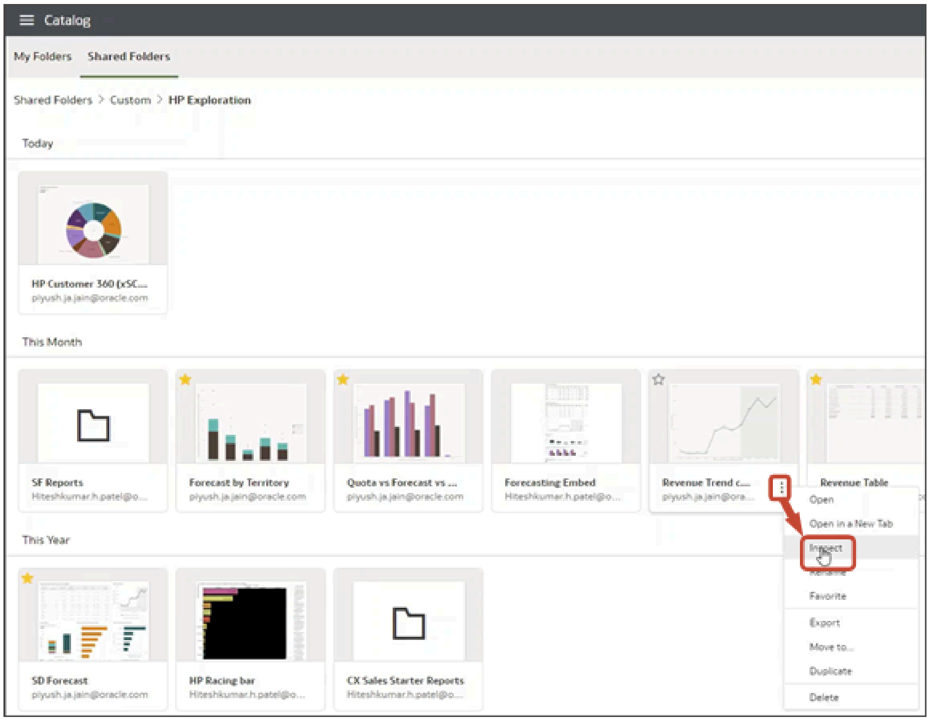
For more information about the Oracle Fusion Data Intelligence embedding framework, see the *Visualizing Data and Building Reports in Oracle Analytics Cloud* guide.

Prerequisites

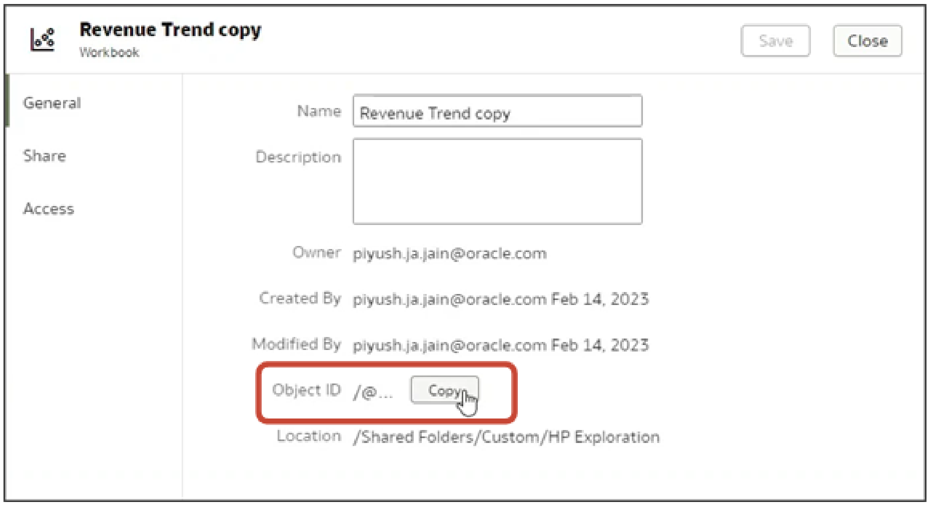
Before adding an Oracle Fusion Data Intelligence (OFDI) component to the Sales Dashboard, you must retrieve the Object ID from your existing OFDI content:

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**.



Name: Revenue Trend copy

Description:

Owner: piyush.ja.jain@oracle.com

Created By: piyush.ja.jain@oracle.com Feb 14, 2023

Modified By: piyush.ja.jain@oracle.com Feb 14, 2023

Object ID: /@...

Location: /Shared Folders/Custom/HP Exploration

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:

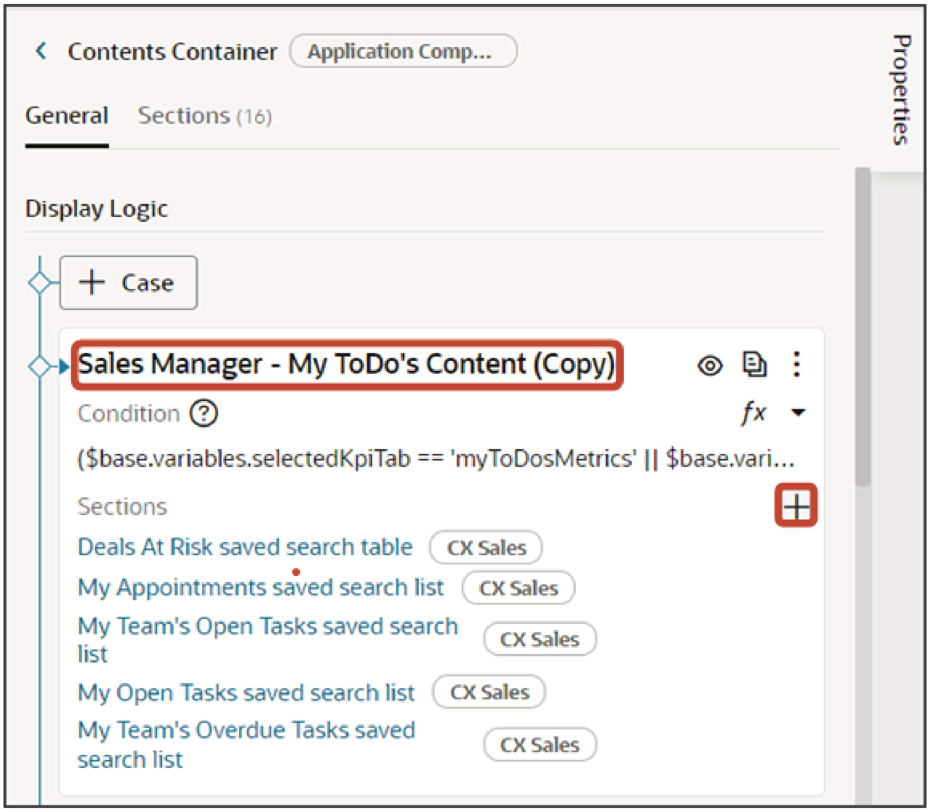
   ```
   https://oac01-gse00010001-px.analytics.ocp.oraclecloud.com/
   ```

   b. Object ID:

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

Add an Oracle Fusion Data Intelligence component to the Sales Dashboard

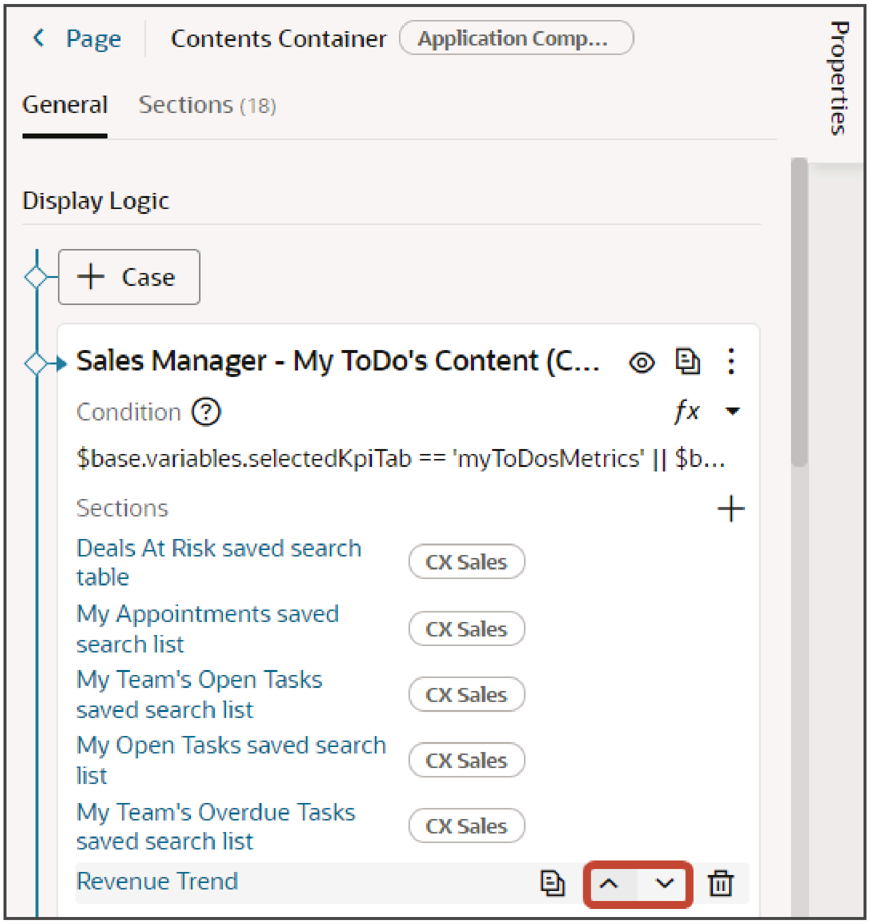
1. On the Properties pane for the **Sales Manager - My ToDo's Content (Copy)** layout, click the **+** Add Section icon.


Contents Container
Application Comp...
General
Sections (16)
Properties
Display Logic
+ Case
Sales Manager - My ToDo's Content (Copy)
Condition ?
fx
($base.variables.selectedKpiTab == 'myToDosMetrics' || $base.vari...
Sections
Deals At Risk saved search table
CX Sales
My Appointments saved search list
CX Sales
My Team's Open Tasks saved search list
CX Sales
My Open Tasks saved search list
CX Sales
My Team's Overdue Tasks saved search list
CX Sales

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.

   < Page | Contents Container (Application Comp...)

   Properties

   General | Sections (18)

   Display Logic

   + Case

   Sales Manager - My ToDo's Content (C...

   Condition ? *fx*

   $base.variables.selectedKpiTab == 'myToDosMetrics' || $b...

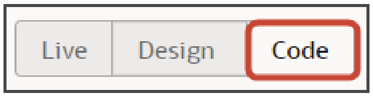
   Sections +

Deals At Risk saved search table	CX Sales
My Appointments saved search list	CX Sales
My Team's Open Tasks saved search list	CX Sales
My Open Tasks saved search list	CX Sales
My Team's Overdue Tasks saved search list	CX Sales
Revenue Trend	

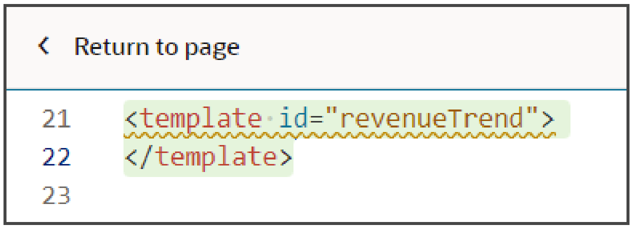
3. Click **Revenue Trend**.

   The template editor opens.

4. Click the **Code** button.

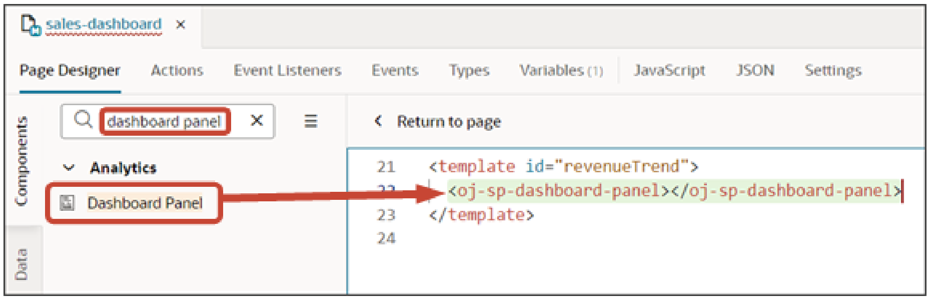


Your new **Revenue Trend** section displays with empty placeholder `template` tags.



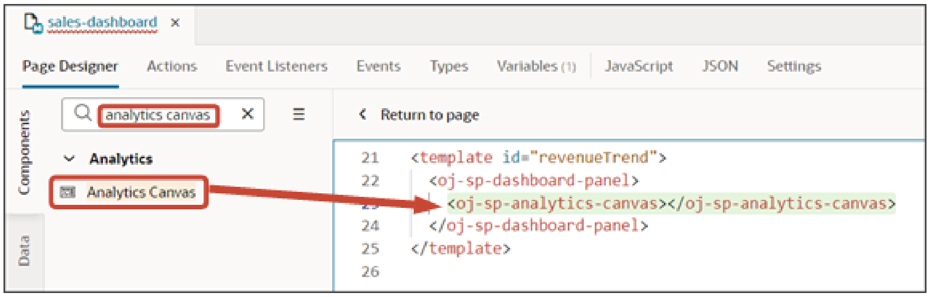
```
<template id="revenueTrend">
</template>
```

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.



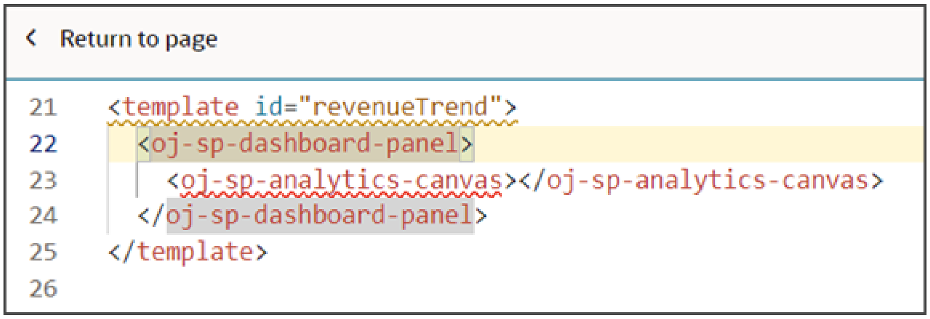
```
<template id="revenueTrend">
  <oj-sp-dashboard-panel></oj-sp-dashboard-panel>
</template>
```

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.



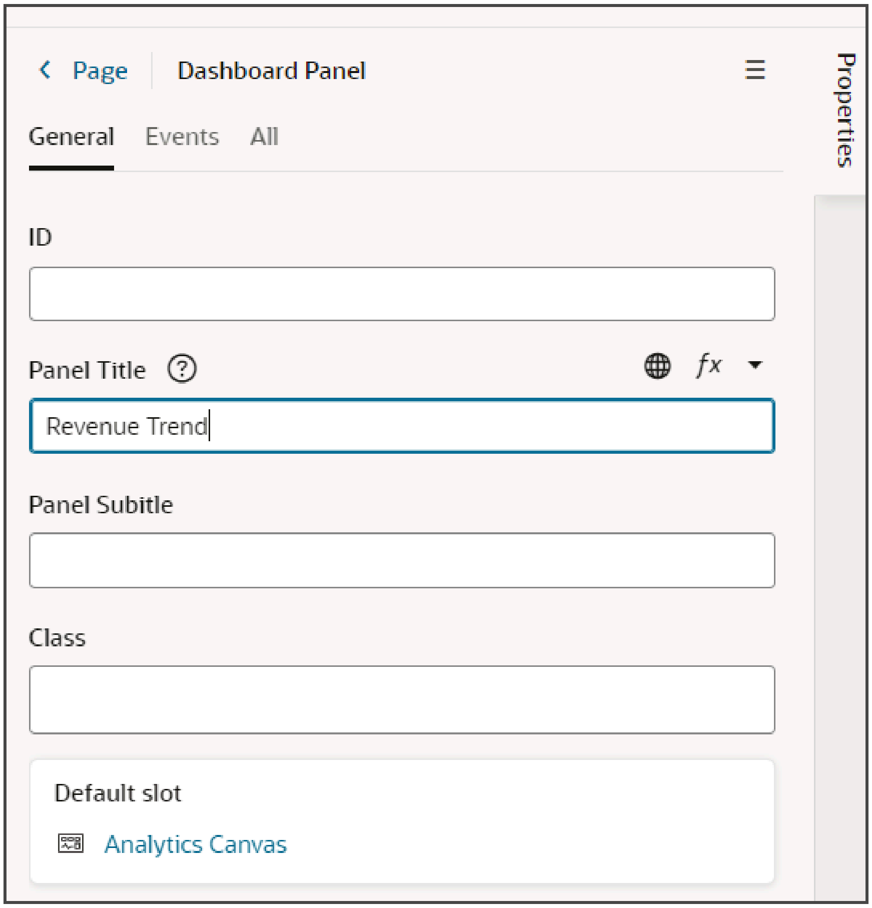
```
<template id="revenueTrend">
  <oj-sp-dashboard-panel>
    <oj-sp-analytics-canvas></oj-sp-analytics-canvas>
  </oj-sp-dashboard-panel>
</template>
```

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



```
<template id="revenueTrend">
  <oj-sp-dashboard-panel>
    <oj-sp-analytics-canvas></oj-sp-analytics-canvas>
  </oj-sp-dashboard-panel>
</template>
```

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`.



ID

Panel Title: Revenue Trend

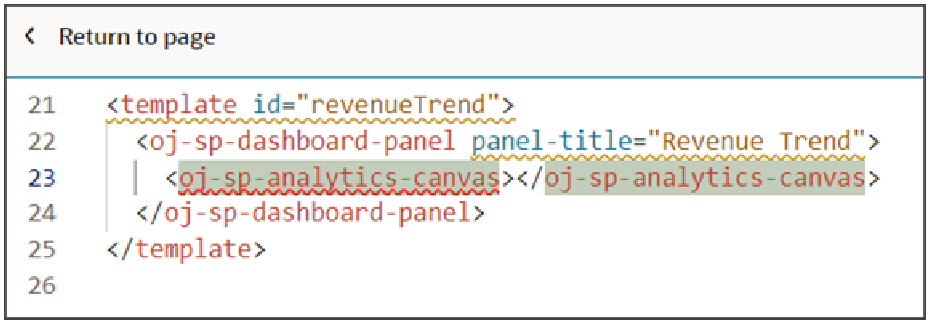
Panel Subitle

Class

Default slot

Analytics Canvas

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

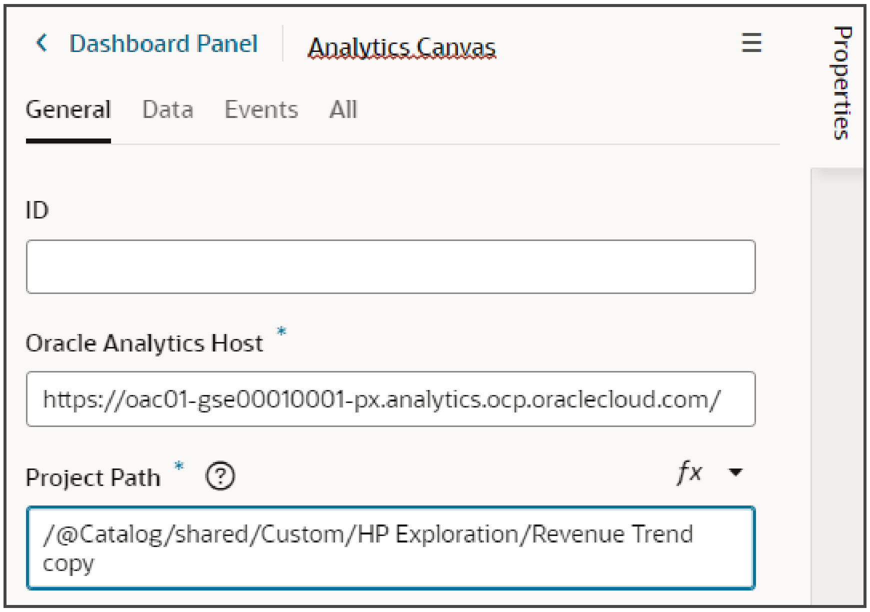


```
<template id="revenueTrend">
  <oj-sp-dashboard-panel panel-title="Revenue Trend">
    <oj-sp-analytics-canvas></oj-sp-analytics-canvas>
  </oj-sp-dashboard-panel>
</template>
```

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.

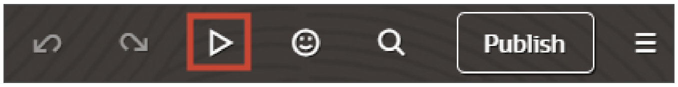
    Oracle Analytics Host *

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

    Project Path *

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

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
    ```

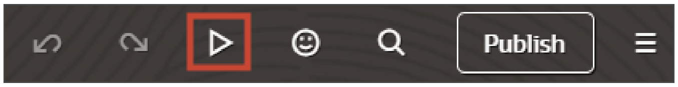
    c. Change the preview link as follows:

    ```
    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:

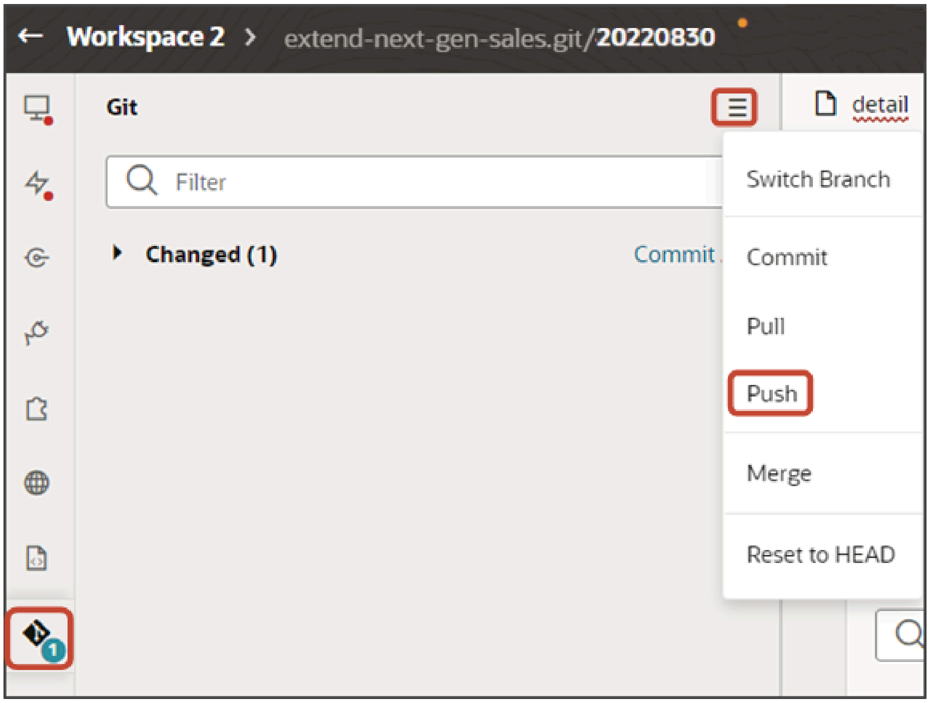
```
https://<servername>/fscmUI/redwood/cx-sales/dashboards/sales-dashboard
```

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.

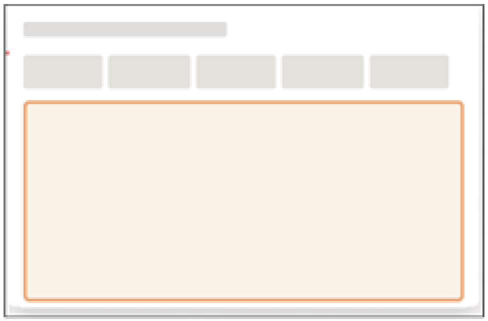
The Default Above-Fold Content Style

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.

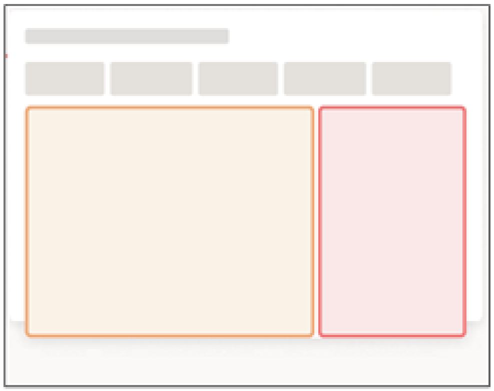
- 1 section

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

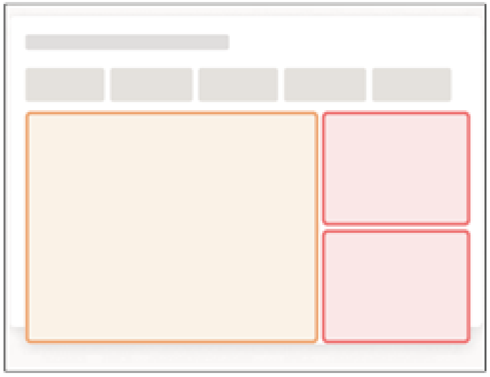
- 2 sections

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

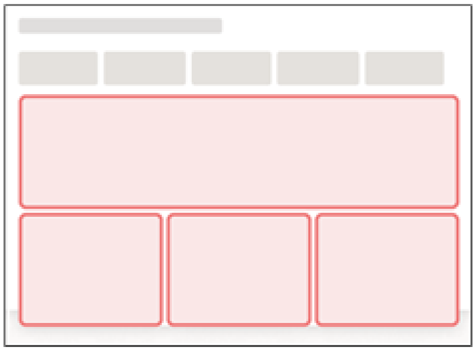
- 3 sections

  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.

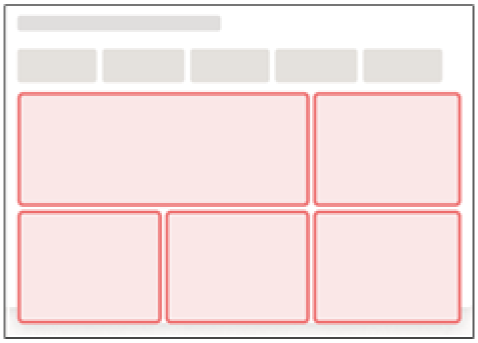
- 4 sections

  The first section takes up the full width of the dashboard but only a 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.

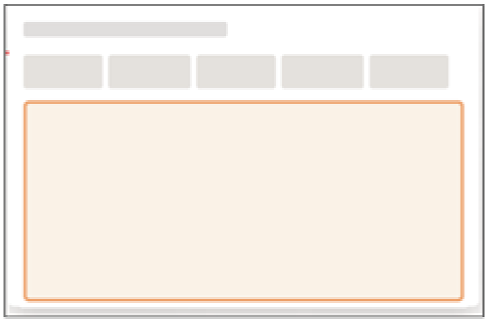
  

The Alternate Hero-plus Layout

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

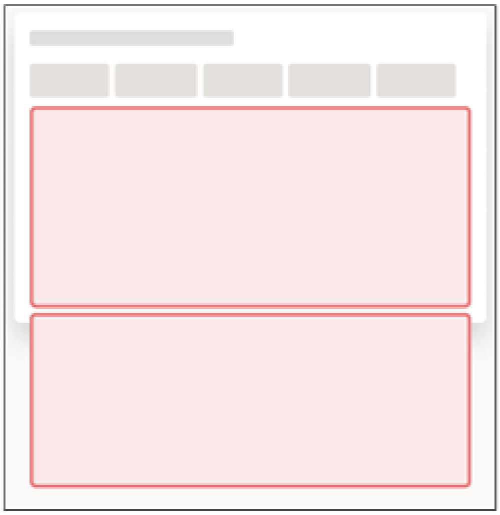
- 1 section

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



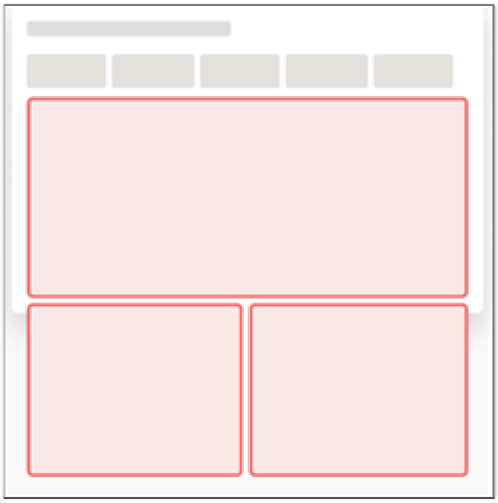
- 2 sections

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



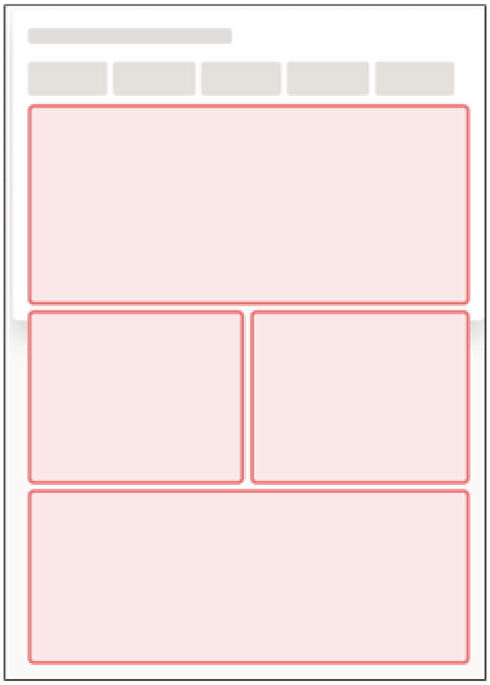
- 3 sections

  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.

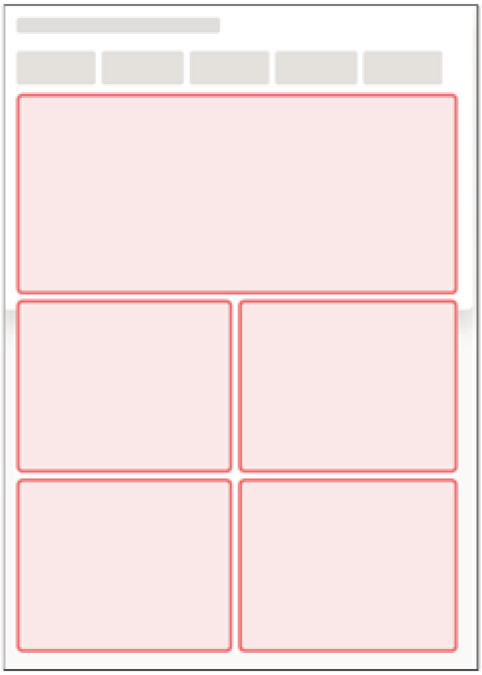
- 4 sections

  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

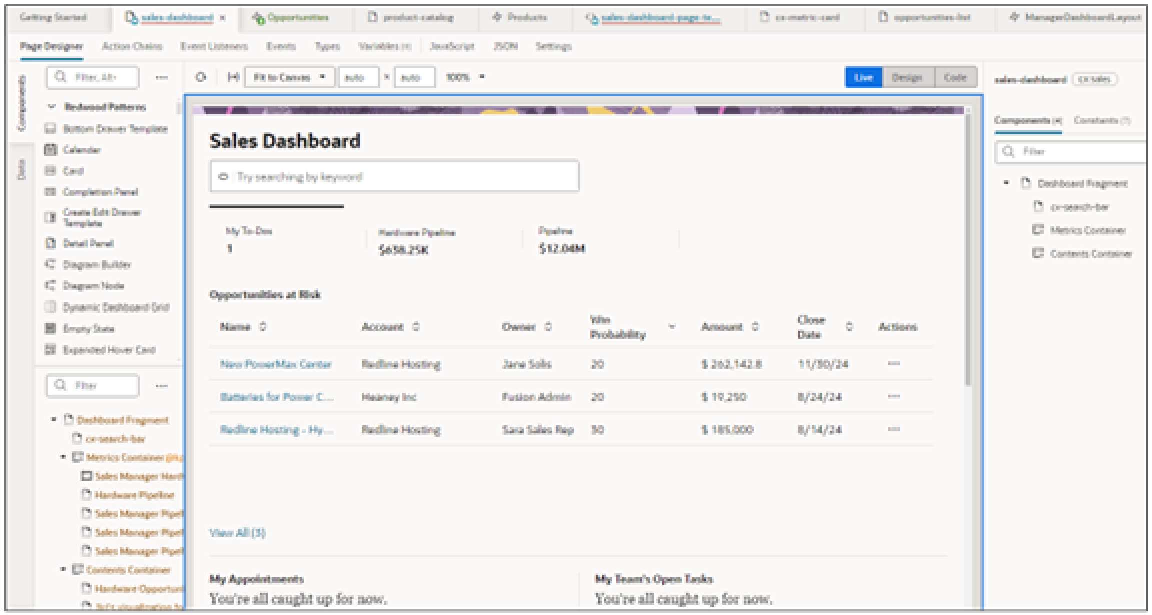
  The first section takes up the full width and height of the dashboard. The rest of the sections appear underneath at half the width.

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.



Sales Dashboard

Try searching by keyword

My To-Dos 1

Hardware Pipeline $658.25K

Pipeline $12.04M

Opportunities at Risk

Name	Account	Owner	Win Probability	Amount	Close Date	Actions
New PowerMax Center	Redline Hosting	Jane Solis	20	$ 262,142.8	11/30/24	···
Batteries for Power C...	Heaney Inc	Fusion Admin	20	$ 19,250	8/24/24	···
Redline Hosting - Hy...	Redline Hosting	Sara Sales Rep	30	$ 185,000	8/14/24	···

View All (3)

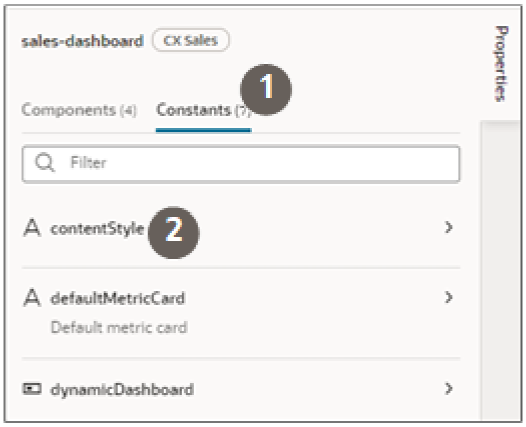
My Appointments

You're all caught up for now.

My Team's Open Tasks

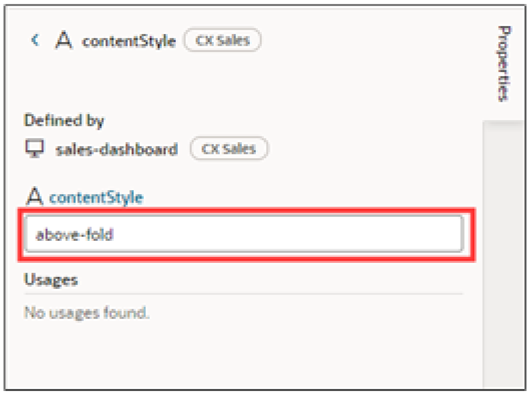
You're all caught up for now.

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.



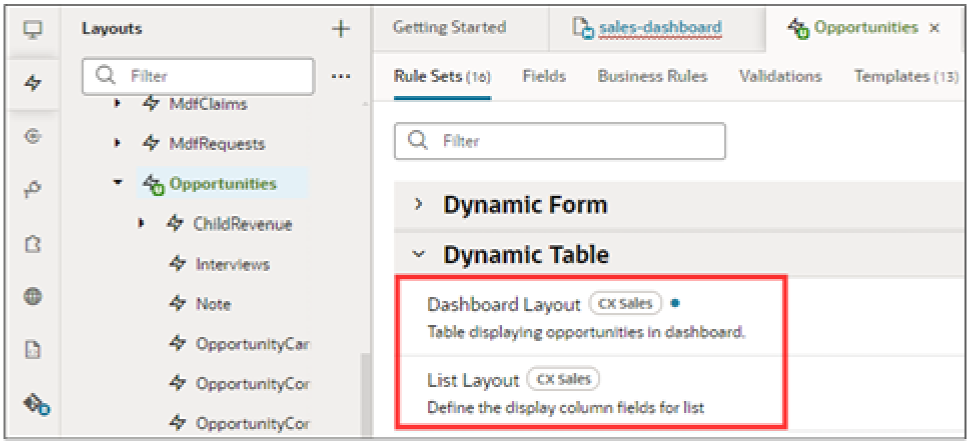
contentStyle: above-fold

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

While what you enter in a 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.

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 on the content container and making a selection from a list.

1. If the copy of the content container you made already includes 5 sections, you must delete a section to add a new one. To delete a section, select a row and click **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

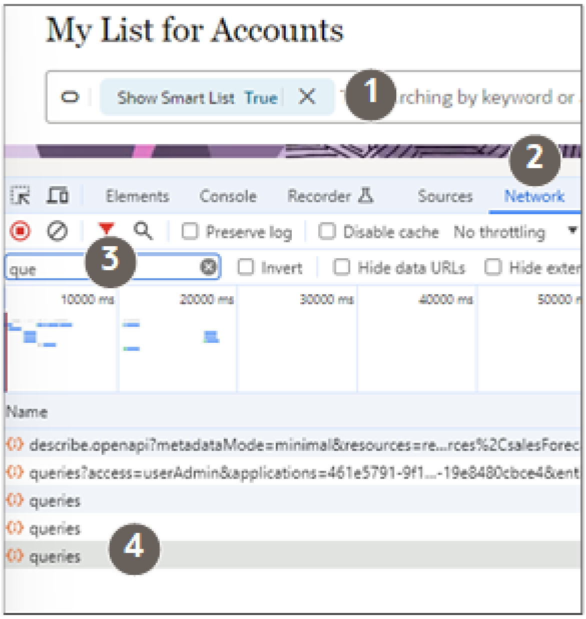
- Lead Amounts by Team Member chart
- My Team's Performance chart

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

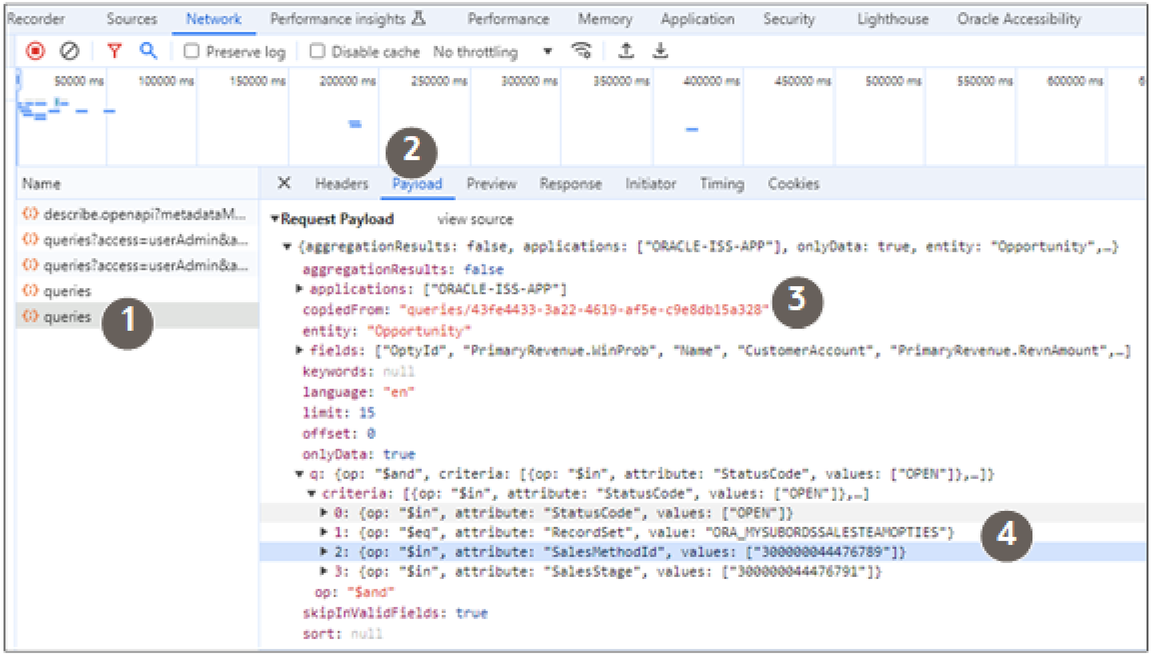
Here's how to find the Universally Unique Identifier (UUID) for a saved search and the various IDs of its query parameters in Oracle Sales in the Redwood User Experience. The IDs are required to reference the saved search and filter parameters in Sales Dashboard layouts, including metric cards and tabular 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).



```
▼Request Payload        view source
 ▼ {aggregationResults: false, applications: ["ORACLE-ISS-APP"], onlyData: true, entity: "Opportunity",…}
    aggregationResults: false
   ▶ applications: ["ORACLE-ISS-APP"]
    copiedFrom: "queries/43fe4433-3a22-4619-af5e-c9e8db15a328"
    entity: "Opportunity"
   ▶ fields: ["OptyId", "PrimaryRevenue.WinProb", "Name", "CustomerAccount", "PrimaryRevenue.RevnAmount",…]
    keywords: null
    language: "en"
    limit: 15
    offset: 0
    onlyData: true
   ▼ q: {op: "$and", criteria: [{op: "$in", attribute: "StatusCode", values: ["OPEN"]},…]}
     ▼ criteria: [{op: "$in", attribute: "StatusCode", values: ["OPEN"]},…]
       ▶ 0: {op: "$in", attribute: "StatusCode", values: ["OPEN"]}
       ▶ 1: {op: "$eq", attribute: "RecordSet", value: "ORA_MYSUBORDSSALESTEAMOPTIES"}
       ▶ 2: {op: "$in", attribute: "SalesMethodId", values: ["300000044476789"]}
       ▶ 3: {op: "$in", attribute: "SalesStage", values: ["300000044476791"]}
      op: "$and"
    skipInValidFields: true
    sort: null
```

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" } ]]
```

5 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 you selected 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 get 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 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, 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 dimension 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 for 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 tabular visualization?

You must change the sort order in the Oracle Transactional Business Intelligence (OTBI) analysis. You can't change the sort order in the visualization itself.

How many rows of data does the tabular visualization show?

The tabular visualization displays the first 100 rows of data using the sort order in the underlying Oracle Transactional Business Intelligence (OTBI) analysis.

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

Check the following:

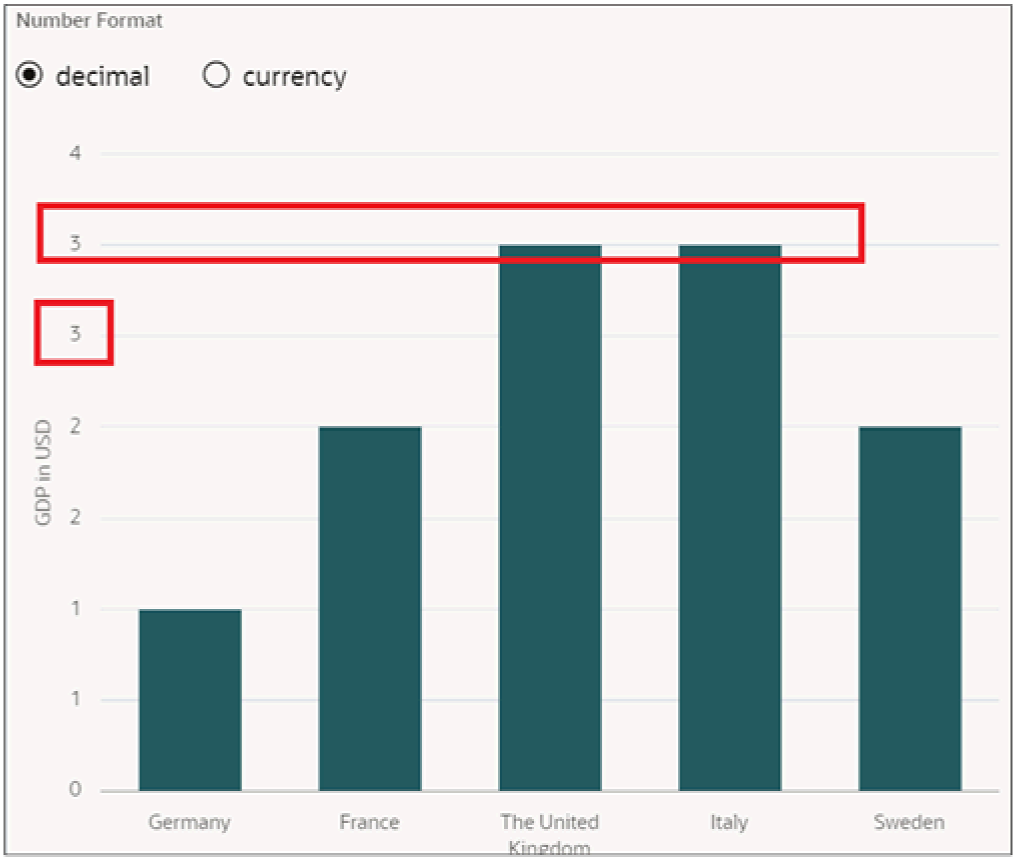
- Can you access the report in Oracle Transactional Business Intelligence (OTBI).
- Is the correct top-level folder selected? If not, change the folder using the **Select Catalog Folder** button.
- New OTBI analyses take a while to appear in the visualization configuration tool.

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 know issue that's in the process of being fixed.



Number Format

● decimal ○ currency

How come I don't see all the rows of my data in a tabular 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.

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 you 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.

6 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:

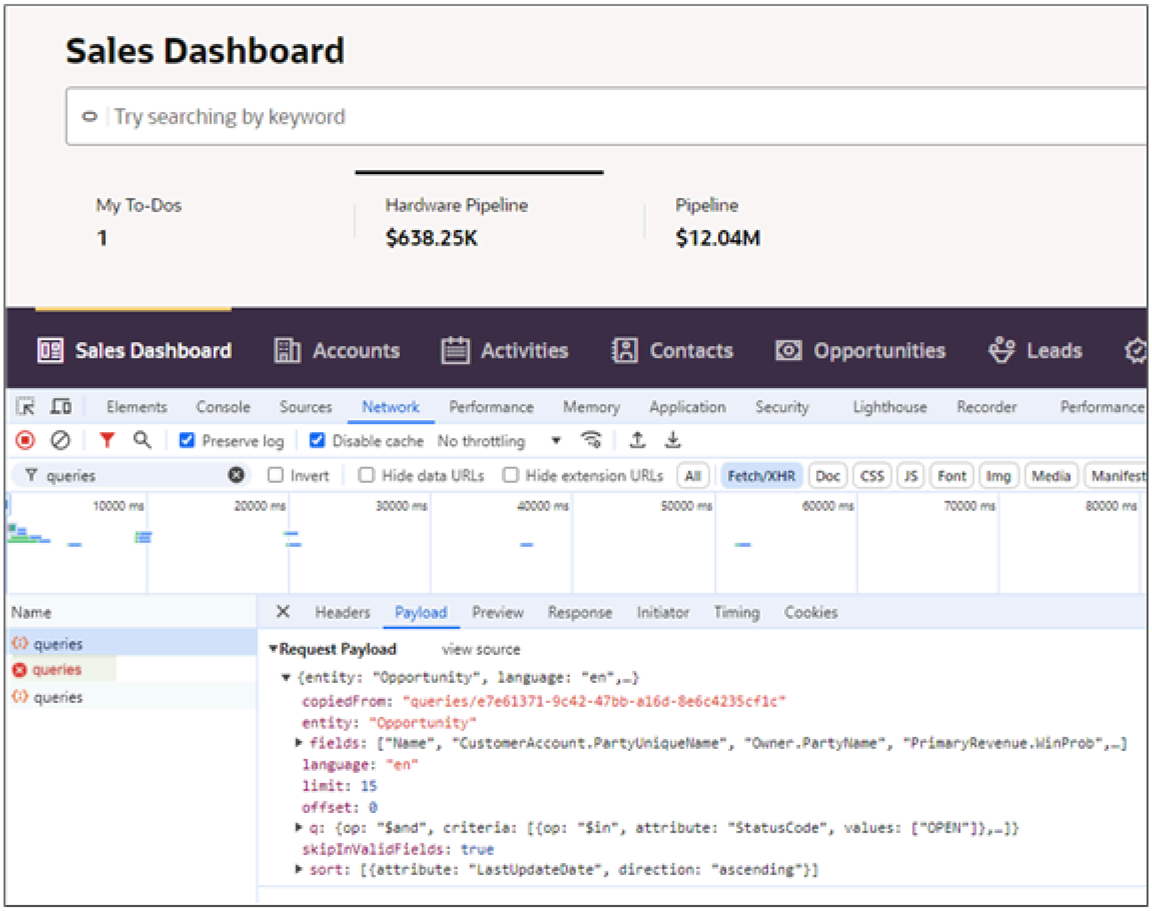
- ORA_ZBS_SALES_REPRESENTATIVE_JOB
- ORA_ZBS_SALES_MANAGER_JOB
- ORA_ZBS_SALES_VP_JOB

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 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**.



My To-Dos
1

Hardware Pipeline
$638.25K

Pipeline
$12.04M

▾Request Payload

```
▾ {entity: "Opportunity", language: "en",…}
   copiedFrom: "queries/e7e61371-9c42-47bb-a16d-8e6c4235cf1c"
   entity: "Opportunity"
  ▸ fields: ["Name", "CustomerAccount.PartyUniqueName", "Owner.PartyName", "PrimaryRevenue.WinProb",…]
   language: "en"
   limit: 15
   offset: 0
  ▸ q: {op: "$and", criteria: [{op: "$in", attribute: "StatusCode", values: ["OPEN"]},…]}
   skipInValidFields: true
  ▸ sort: [{attribute: "LastUpdateDate", direction: "ascending"}]
```

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 right 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"