



# Oracle CX Sales Integration App

## User Guide

# Contents

---

|   |           |
|---|-----------|
| <b>Oracle CX Sales Integration app</b>                            | <b>4</b>  |
| Supported Oracle CX Sales objects                                 | 5         |
| <b>Getting started with the Oracle CX Sales app</b>               | <b>7</b>  |
| Creating recommended contact and account fields                   | 9         |
| Installing the Oracle CX Sales Integration app                    | 11        |
| Configuring connections   | 12        |
| Modifying a connection  | 16        |
| Creating default assets   | 17        |
| Migrating from the native integration                             | 20        |
| Creating step actions   | 23        |
| Creating campaign actions   | 27        |
| Creating response actions   | 31        |
| Creating imports  | 34        |
| Viewing and editing imports                                       | 39        |
| Scheduling an import  | 41        |
| Adding imports to a canvas  | 42        |
| Creating campaign imports   | 45        |
| Creating opportunity imports                                      | 49        |
| Creating user imports   | 54        |
| Configuring notifications   | 58        |
| <b>Using the Oracle CX Sales app in a program or campaign</b>     | <b>61</b> |
| <b>Viewing report data in the Oracle CX Sales Integration app</b> | <b>64</b> |
| Viewing the global report dashboard in a cloud menu               | 64        |
| Viewing report data for actions and imports                       | 68        |
| Action report data  | 68        |

---

|   |           |
|---|-----------|
| Import report data .....                          | 70        |
| <b>Example programs .....</b>                     | <b>73</b> |
| Creating a create unique program .....            | 74        |
| Creating a hybrid program .....                   | 78        |
| Creating a point of interest program .....        | 80        |
| Create an Opportunity Qualification Program ..... | 83        |
| CRM Integration program overview .....            | 84        |
| CRM Integration program setup .....               | 86        |
| Campaign Lead data in Eloqua .....                | 96        |

# Oracle CX Sales Integration app

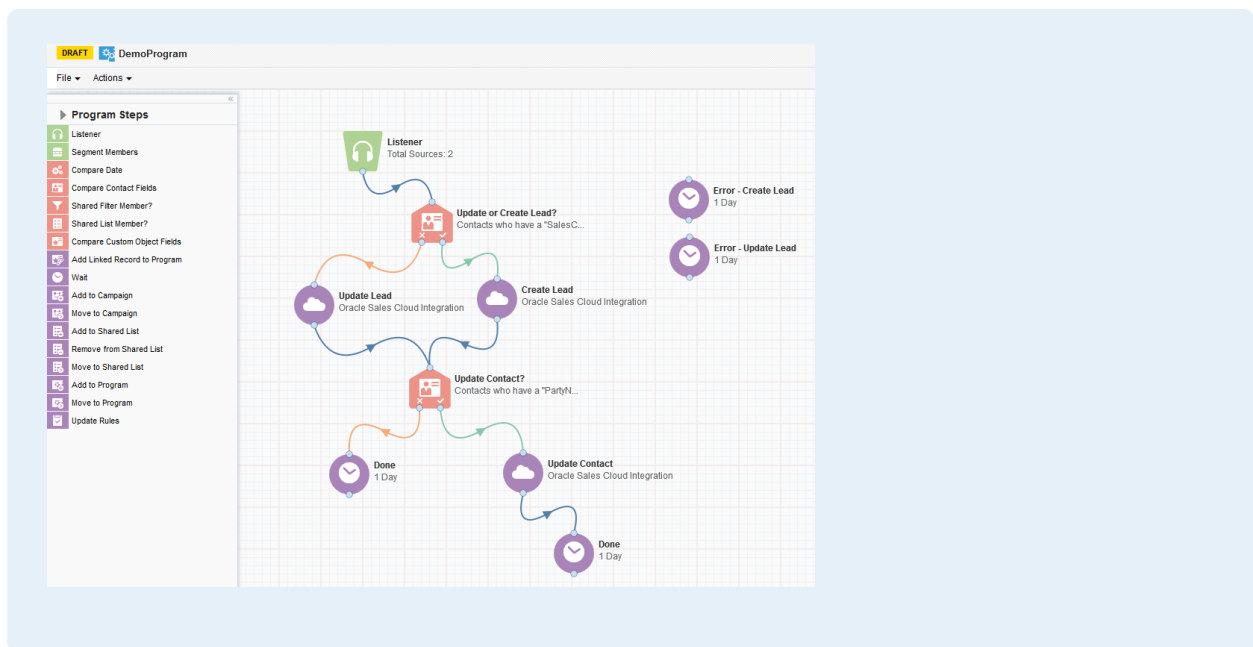
**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

The Oracle CX Sales Integration app creates an integration between Oracle Eloqua and Oracle CX Sales to send data between the two platforms. Leverage the Oracle CX Sales Integration app in your marketing cloud solution to:

- Transfer the leads generated from marketing campaigns in Oracle Eloqua into Oracle CX Sales so they are nurtured and synchronized as sales leads.
- Use the app within a program or campaign canvas to create or update records in supported Oracle CX Sales objects in real time. Typical use cases include lead creation, lead updates, and contact updates.
- Schedule imports from Oracle CX Sales to Oracle Eloqua for [supported Oracle CX Sales objects](#).

There are various use cases where the app could be used for marketing qualification processes and lead generation.

**Example:** A listener step could be configured to listen for contacts who complete a form. Once the contact is added to the program canvas, you may want to decide to create a lead for the contact, depending on the next decision, you can configure this app to either create a new lead or update an existing lead within Oracle Eloqua.



You can manage user permissions on this app by following the instructions listed [here](#).

## Supported Oracle CX Sales objects

The integration supports bi-directional data flow for the objects listed below. This means data may be transferred both to and from Oracle Eloqua or Oracle CX Sales for these objects. The typical use of this app is to import data in these objects to Eloqua, and to create leads and update contacts in Oracle CX Sales. Currently, the Oracle CX Sales Integration app supports the following objects:

- Activity
- Campaign
- Campaign member
- Deal registration
- Opportunity

- Partner
- Partner Contact
- Products
- CX Sales account SDO
- CX Sales contact SDO
- CX Sales household SDO
- Sales lead
- Top level custom objects






Download the [Oracle CX Sales Integration App User Guide](#).

# Getting started with the Oracle CX Sales app

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

The steps below outline the recommended setup and configuration process for the Oracle CX Sales Integration app.

1. Prior to installing the app, [create contact and account fields](#) which will be used to map Eloqua fields to Oracle CX Sales fields when configuring and using the app. These fields will be used in the Connections area to specify which Eloqua fields are to be used for matching with Oracle CX Sales fields.
2. [Install](#) the Oracle CX Sales Integration app. Once the app is installed, configuration settings for [connections](#), [actions](#), [imports](#), and [notifications](#) can be found in the apps list. Go to **Settings**  > **Apps** and click **Configure**  to access the configuration settings.
3. In Eloqua, click the *cloud menu*  icon on the right hand side to expand the *Apps* list and select **Oracle CX Sales Integration**. The *App Configuration for Oracle CX Sales Integration* page is displayed.
4. Configure your CRM [connections](#). Each CRM system can have multiple connections, including alternate credentials or sandbox configurations. Only one connection can be active per CRM system.
5. If you have an existing native Oracle CX Sales integration, you can [migrate its assets](#) to the Oracle CX Sales Integration app.

6. [Create default assets](#) for the connection to automatically create actions and imports.

Alternatively, you can create actions and imports by manually configuring them as described in the following steps.

7. [Create actions](#) (data transactions) between Oracle Eloqua and Oracle CX Sales.

- Use [step actions](#) to create and update records to be used in a program or campaign.
- Use [campaign actions](#) to send campaign updates from an Oracle Eloqua campaign object to a custom object in Oracle CX Sales. One campaign update action can be active at a time.
- Use [response actions](#) to send data to Oracle CX Sales based on response rules configured in your Oracle Eloqua instance. When a response occurs, Oracle Eloqua records the response based on priority and the response is sent to Oracle CX Sales.

**Important:** The response actions feature is currently released under our controlled availability program. To request access to this feature, please log in to [My Oracle Support](#) and create a service request.

8. Create and schedule imports to import data from Oracle CX Sales to Oracle Eloqua periodically. The Oracle CX Sales Integration app executes imports every 15 minutes. If an import needs to be performed off schedule, it can be executed on demand.

- Use [standard imports](#) to import contact record data from Oracle CX Sales into Oracle Eloqua. The following imports are recommended: get leads, get contacts, get accounts.
- Use [campaign imports](#) to import custom object data from Oracle CX Sales into Oracle Eloqua campaigns to allow for bi-directional updates between the two applications.

9. Setup email [notifications](#) to be alerted if there are problems with your imports or actions.

10. Review your [reports](#). The Oracle CX Sales Integration app provides reporting on the records being synced between Eloqua and Oracle CX Sales to provide insight into how your syncs are performing.

11. [Use the app in a campaign or program](#) to update or create contact records. View recommended programs [here](#).



12. Add imports to a canvas to run up to ten post processing actions.

## Creating recommended contact and account fields

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

Before installing the Oracle CX Sales Integration app, we recommend that you first create contact and account fields which will be used to map Oracle Eloqua fields to Oracle CX Sales fields when configuring and using the app. These fields will be used in the Connections area to specify which Oracle Eloqua fields are to be used for matching with Oracle CX Sales fields.

### To create the recommended fields:

1. Create contact fields and account fields. The following table outlines how these fields will be used in the app and provides suggested names for each field.

| Eloqua Object | Eloqua Field Name*                   | Field Purpose (Oracle CX Sales ID)                   |
|---------------|--------------------------------------|--|
| Contact       | <b>OSC - Lead Identifier</b>         | <b>Lead Identifier (LeadId)</b>                      |
|               | <b>OSC - Contact Registry ID</b>     | <b>Registry ID (PartyNumber)</b>                     |
|               | OSC - Contact Party ID               | Party ID (PartyId)                                   |
|               | OSC - Account Registry ID            | Related Account Party Number<br>(AccountPartyNumber) |
|               | OSC - Account Party ID               | Related Account Party Id (AccountPartyId)            |
|               | OSC - Contact Primary Address Number | Site Number (PrimaryAddress.AddressNumber)           |
|               |                                      |  |
| Account       | <b>OSC - Account Registry ID</b>     | <b>Registry ID (PartyNumber)</b>                     |
|               | OSC - Account Party ID               | Party ID (PartyId)                                   |

*\*Suggested field name*

The fields in bold in the table above are to be used in the Connections area to specify which Oracle Eloqua fields are to be used for storing Oracle CX Sales identifiers. The values in these fields are used to identify existing records in Oracle CX Sales when performing an update, and to store the ID value from Oracle CX Sales when performing a create.

**Important:** Account linkage in Oracle Eloqua should be configured to use the Account Registry ID/Party Number field on both the account and contact. Any account import in the integration must be set up to uniquely match on the account field configured in account linkage.

# Installing the Oracle CX Sales Integration app

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

**Important:** Because deleting or deactivating the installer user account will impact installed apps, we recommend installing apps using a user account that is not tied to a specific person. The user account must have customer administrator rights.


## To install the Oracle CX Sales Integration app:

1. Follow the URL provided:





<https://cloudmarketplace.oracle.com/marketplace/app/AppOracleCXSales>

**Note:** If you have IP allowlists enabled, [add internal Eloqua IPs](#) to your allowlist .

2. Log in with your Eloqua credentials, if required.
3. Click **Accept and Install** to add the app to your *Apps* list.
4. Click **Sign In**.
5. Confirm your credentials and click **Accept** to grant permission for the app to communicate with Oracle Eloqua on your behalf.

The app is now installed and can be viewed in your [Apps](#) list (**Settings**  > **Apps**, under the *Platform Extensions* section). You can [configure](#) your app now or at a later time.


You can return to the *Apps* list at any time and view the app description, modify the configuration settings, reinstall, or uninstall it. You can also check its status and dependencies.

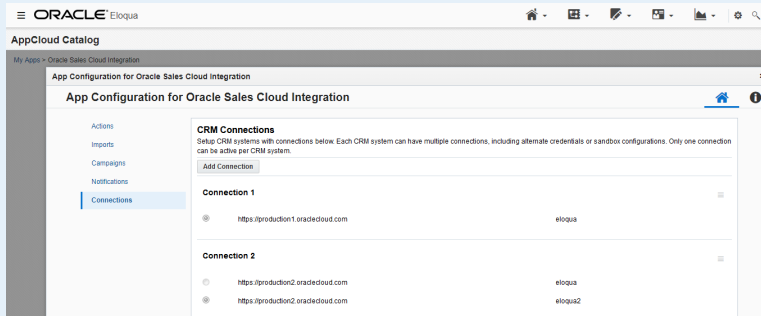
 **Warning:** To modify configuration settings, navigate to *Apps*, select your app, and click the **Configure** icon . Selecting the **Reinstall** icon  will force you to go through the entire configuration process again. The app will be unavailable for all users during the reinstall. All existing assets, configurations and history are preserved when you reinstall an app. Also, you can choose to delete an app by clicking the **Uninstall** . In this case, all current assets, configurations, and history are permanently deleted.

## Configuring connections


**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

The *Connections* page in the Oracle CX Sales Integration app lists all of your CRM connections and associated credentials.

 **Example:** You may want to have two sets of connections. One connection may have one set of credentials, while another connection has two credentials (one set of credentials could be linked to a sandbox or test environment and the other could be production credentials).



## To configure connections:

1. In Eloqua, click the *cloud menu*  icon on the right hand side to expand it. This menu is located on the My Eloqua Dashboard, landing page editor, email editor, and campaign canvas.
2. From the *Apps* list, click **Oracle CX Sales Integration**. The *App Configuration for Oracle CX Sales Integration* page is displayed.
3. In the Oracle CX Sales Integration app, click **Connections**.
4. Click **Add Connection**.

## 5. Complete the connection details:


Name Connection 1

Credentials


Add Credential

URL https://production1.oraclecloud.com

Username eloqua

Password \*\*\*\*\* 

Status Validate Credentials

Active Credential  Use this credential for the active connection

Default Field Mappings

|          | Oracle Sales Cloud  | Eloqua Fields             |
|----------|---------------------|---------------------------|
| Leads    | Identifier (LeadId) | OSC - Lead Identifier x ▾ |
|          | LeadID              | C_OSC__Lead_Identifier1   |
| Contacts | Registry ID         | OSC - Registry ID x ▾     |
|          | PartyNumber         | C_OSC__Registry_ID1       |
| Accounts | Registry ID         | OSC - Registry ID x ▾     |


- Name:** Enter a unique name for the connection. Connections cannot be renamed. The connection name will be included in the name of any [default actions, imports, and programs](#).
- Add one or more credentials:
  - URL:** Enter the URL for the CRM connection.
  - Username:** Enter user name credentials for the connection.
  - Password:** Enter password credentials for the connection. Click **Show Password** to display the characters in the **Password** field. Click **Validate Credentials** to verify that proper access to Oracle CX Sales is available. Click the resulting **Valid** or **Invalid** status message to view details.

### Connection Validation

#### Success

This connection was able to successfully connect to CRM. Please click close to continue with the installation process.

 Lead Object Found

 Contact Object Found

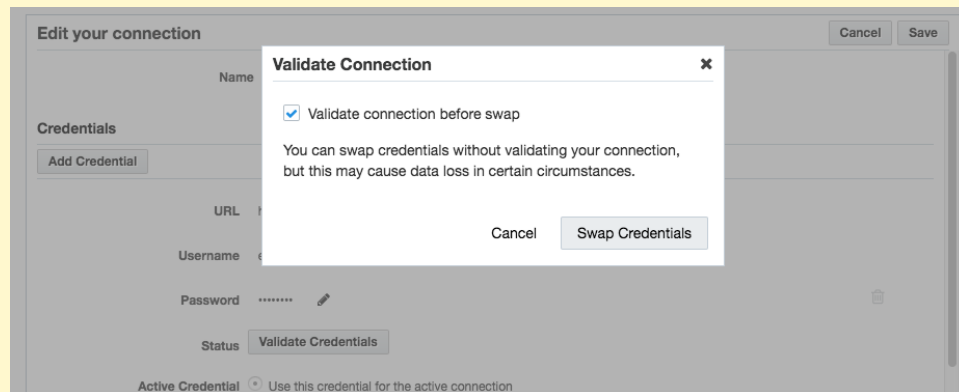
 Account Object Found

 Opportunity Object Found

**Note:** In order for a credential to be added, deleted, or set to default, the connection must first be saved for the change to persist.

- c. Activate one credential for the connection by selecting the **Active Credential** radio button.

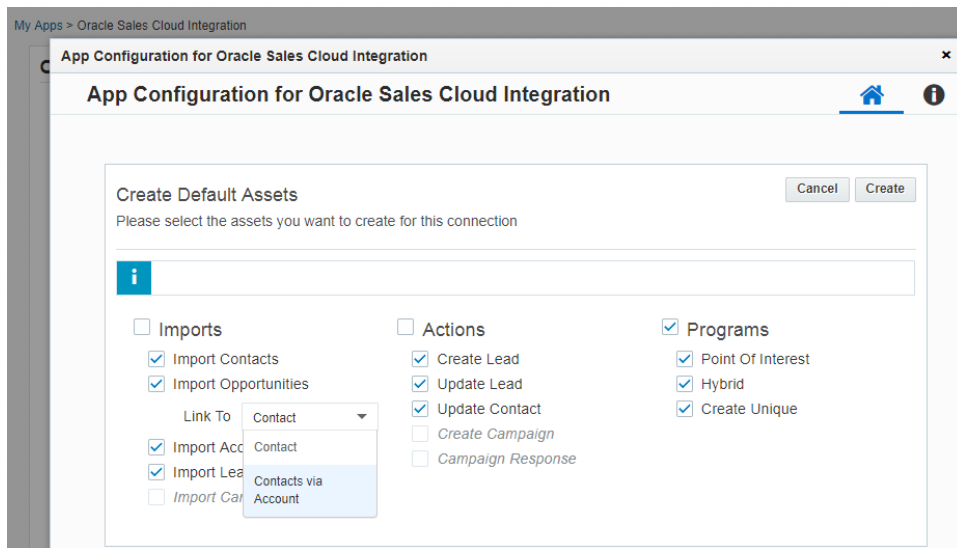
**Important:** When editing your connection and activating a different credential, you are given an option whether to validate credentials before swapping. The validation option is recommended and selected by default. If the validation option is not selected, the credential swap occurs instantly and the imports and actions are not disabled, which may result in action and import errors.




- d. **Default Field Mappings:** Select the Eloqua fields to map to the corresponding CX Sales Unique Identifier fields.

- **Leads:** Select an Oracle Eloqua field to map to the Lead Identifier field for Oracle CX Sales contacts.
- **Contacts:** Select an Oracle Eloqua field to map to the Registry ID field for Oracle CX Sales contacts.
- **Accounts:** Select an Oracle Eloqua field to map to the Registry ID field for Oracle CX Sales accounts.


6. Click **Save**. The *Create Default Assets* page is displayed.



7. Select the actions, imports, and programs that you want to create. For more details, see [creating default assets](#).
8. Click **Create**. Selected default imports, actions, and programs are created with a name that includes the name of your connection. The imports are created in the disabled state and actions are created in the enabled state. To enable or disable them, go to the [Imports](#) or [Actions](#) tab and click the corresponding menu icon . If you selected default programs, a message displays the folder where you can find them.

## Modifying a connection

### To modify an existing connection:

1. In the Oracle CX Sales Integration app, click the **Connections** tab.
2. Click the connection's menu icon  and select any of the following options:
  - **Create Action:** Click to [create an action](#) and automatically associate it with the current connection.
  - **Create Import:** Click to [create an import](#) and automatically associate it with the current connection.



- **Edit:** Open the connection in Edit mode so that you can add, modify, or delete a credential and modify its field mappings. You cannot modify the name of a credential.
- **Create Default Assets:** [Create default imports and actions](#) if you did not already do this as part of the connection set up.
- **Migrate Native Assets:** [Run the migration wizard](#) to automatically migrate assets from your native Oracle CX Sales integration.
- **Add Credential:** Open the connection in Edit mode so that you can add, modify, or delete a credential and modify its field mappings.
- **Delete:** You can delete connections if there are no associated actions or imports. A confirmation dialog will list any dependencies. The dependent actions and imports need to be disabled and deleted before deleting the connection.




3. If a connection has multiple credentials, you can select the radio button to set the default credential. Only one credential per connection can be active at once.

## Creating default assets

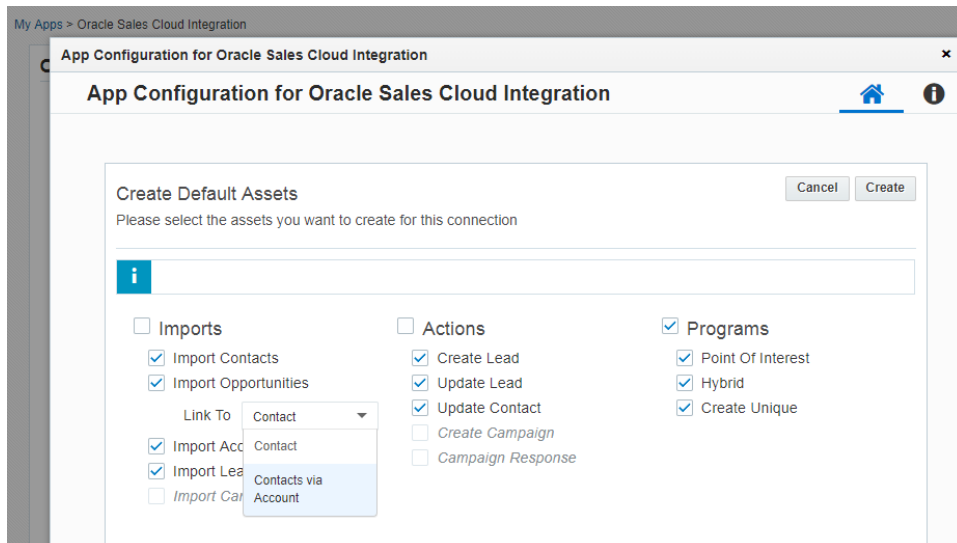
**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

Once you [configure a connection](#), the *Create Default Assets* page offers to automatically create default actions, imports, and programs using default mappings. You can alternatively create [actions](#), [programs](#), and [imports](#) by manually configuring them.

## To create default assets:

1. In the Oracle CX Sales Integration app, click **Connections**.
2. Do one of the following:
  - For a new connection, click **Add Connection**, [configure a connection](#), and click **Save**.
  - For an existing connection, click the connection's menu icon  and select **Create Default Assets**.

The *Create Default Assets* page is displayed.



My Apps > Oracle Sales Cloud Integration

App Configuration for Oracle Sales Cloud Integration

Create Default Assets

Please select the assets you want to create for this connection

☐ Imports

☒ Import Contacts

☒ Import Opportunities

Link To:

☒ Import Accounts

☒ Import Leads

☐ Import Campaigns

☐ Actions

☒ Create Lead

☒ Update Lead

☒ Update Contact

☐ Create Campaign

☐ Campaign Response

☒ Programs


☒ Point Of Interest


☒ Hybrid

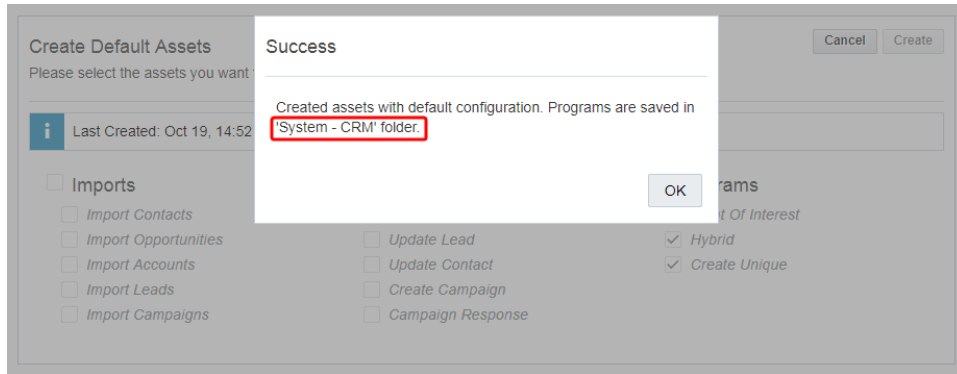
☒ Create Unique

3. Select any of the imports, actions, and programs that you want to create.
  - **Imports:** Selects all of the following [imports](#).
    - **Import Contacts**
    - **Import Opportunities:** If you select this option, you can optionally [link opportunities](#) to contacts directly or to contacts via accounts.
    - **Import Accounts**

- **Import Leads**
- **Import Campaigns**
- **Actions:** Selects all of the following [actions](#).
  - **Create Lead**
  - **Update Lead**
  - **Update Contact**
  - **Create Campaign**
  - **Campaign Response**
- **Programs:** Selects all of the following to create a [program](#) canvas for each:
  - **Point Of Interest**
  - **Hybrid**
  - **Create Unique**

 **Note:** If an object is not enabled in your Oracle CX Sales account, you cannot select it on the *Create Default Assets* page. For example, support for Oracle CX Sales campaign imports and actions is currently released under our Controlled Availability program. To request access to this feature, please log in to [My Oracle Support](https://support.oracle.com) (<https://support.oracle.com>) and create a service request.

4. Click **Create**. The default actions, imports, and programs are created with a name that includes the name of your connection. The imports are created in the disabled state and actions are created in the enabled state. To enable or disable imports or actions, go to the [Imports](#) or [Actions](#) tab and click the corresponding menu icon . If you selected any programs, a message displays the program folder where you can find your programs, which have not yet been activated.



5. Once the default actions, imports, and programs are created for the connection, you can use them in a [program or campaign](#).

**Note:** You will create duplicate actions, imports, and programs if you select **Create Default Assets** for the same connection, select the same assets, and click **Create**. The duplicate assets will have the same names incremented with integers in the following pattern: `connection_name - asset_name integer`, such as `Production - Hybrid 2`.

## Migrating from the native integration

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

You can run the migration wizard to automatically migrate the following assets from your native Oracle CX Sales integration:


- Accounts
- Contacts
- Leads

For cases where external calls in the native integration include Eloqua custom contact and account fields in their mappings or that use CDOs, you can modify the mappings after completing the migration from your native integration.

## Prerequisites

- An existing [native Oracle CX Sales integration](#) with Oracle Eloqua
- [A configured connection](#)

## To migrate native integration assets:

1. In the Oracle CX Sales Integration app, click **Connections**.
2. Click the connection's menu icon  and select **Migrate Native Assets**. After fetching native assets for the connection, the app displays the *Migrate Assets* page.

App Configuration for Oracle Sales Cloud Integration

Actions

Imports

Campaigns

Notifications

Connections

Migrate Assets

Migrate the below available assets from native integration to Oracle Sales Cloud app.

If the migration was done earlier, the previously migrated assets will be duplicated and not overwritten.

Cancel

Migrate

!




Some assets from native integration without instances in Sales Cloud are not available for migration and disabled for selection.

| <input type="checkbox"/> | Asset Name      | Asset Type              | From Source           | To Destination        |
|--------------------------|-----------------|-------------------------|-----------------------|-----------------------|
| <input type="checkbox"/> | Get Leads       | External Call / Inbound | Eloqua: Leads         | Sales Cloud: Leads    |
| <input type="checkbox"/> | Get Accounts    | External Call / Inbound | Eloqua: Accounts      | Sales Cloud: Accounts |
| <input type="checkbox"/> | Get Contacts    | External Call / Inbound | Eloqua: Contacts      | Sales Cloud: Contacts |
| <input type="checkbox"/> | Create Leads    | Auto Sync / Outbound    | Sales Cloud: Leads    | Eloqua: Leads         |
| <input type="checkbox"/> | Create Accounts | Auto Sync / Outbound    | Sales Cloud: Accounts | Eloqua: Accounts      |
| <input type="checkbox"/> | Create Contacts | Auto Sync / Outbound    | Sales Cloud: Contacts | Eloqua: Contacts      |

3. Select any of the assets that you want to migrate.

**Note:** If an asset in your native integration does not have a corresponding asset in your Oracle CX Sales account, it cannot be selected on the *Migrate Assets* page.

4. Click **Migrate**. The selected assets are migrated with default names. Once the migration is complete, the *Migrate Assets* page displays the migration status for each asset:

-  Indicates that the migration of the asset was successful.
-  Indicates that some records could not be migrated because of a mismatch with the native integration. For example, this warning may indicate that internal or external calls in the native integration were migrated but some field mappings were not migrated. You can manually correct these by editing the corresponding imports and actions in the Oracle CX Sales Integration app.
-  Indicates that the migration of one or more assets failed because the connection to the server could not be made at this time.

You can rerun the migration again at a later time if needed to correct issues due to a failed connection or mismatched records. To correct errors, such as mismatched values, you can manually fix the corresponding imports and actions. If you rerun the migration for successfully migrated assets, they will be duplicated and not overwritten.

5. Once you review the status of each migrated asset, click **Done**. The *Connections* page is displayed.

**Note:** You will create duplicate assets if you select **Migrate Assets** for the same connection, select the same assets, and click **Migrate**. The duplicate assets will have the same names incremented with integers.

# Creating step actions

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

**Note:** This page outlines how to create step actions in the Oracle CX Sales Integration app. You can also create [campaign actions](#), [create response actions](#), or [automatically create actions](#).

Actions are data transactions between Oracle Eloqua and Oracle CX Sales (formerly Oracle Sales Cloud). The *Step Actions* page is where you create the types of actions to be used in a program or campaign when using the app. You can create the following types of actions:

- *Create:* Create new records in Oracle Sales.
- *Update:* Take existing Oracle Eloqua contacts and update their information in Oracle Sales.

## Prerequisites


- A [configured connection](#)

**To create a step action:**

1. In the Oracle CX Sales Integration app, click **Actions**.
2. Click **Add Action** and select a connection for which you want to create a new action.
3. Complete the action details:


The screenshot shows the 'Create Action' dialog box. At the top, there are three tabs: 'Name', 'Action Type', and 'Connection'. The 'Name' tab is selected, showing a text input field with the value 'New Action'. Below the tabs, there are three sections: 'Object Mapping' with dropdowns for 'Eloqua Object' (set to 'Contact') and 'Sales Cloud Object' (set to 'Lead'); 'Update Oracle Sales Cloud Fields' with a table mapping fields from Eloqua to Oracle Sales Cloud; and 'Update Eloqua Fields' with a table mapping fields from Oracle Sales Cloud to Eloqua. The 'Update Oracle Sales Cloud Fields' table has columns 'From Eloqua' and 'To Oracle Sales Cloud'. It contains four rows of mappings: Email Address (C\_EmailAddress) to Contact E-Mail (PrimaryContactEmailAddress), First and Last Name (C\_FirstAndLastName) to Primary Contact (PrimaryContactPartyName), Country (C\_Country) to Country (Country\_o), and Static Text to MapEloquaStaticFieldHere. There are 'Add Export Field Mapping' and 'Add Import Field Mapping' buttons. The 'Update Eloqua Fields' table has columns 'From Oracle Sales Cloud' and 'To Eloqua'. It contains one row: Lead Identifier (LeadId) to SC\_LeadId (C\_SC\_LeadId1), which is marked as a 'Unique Identifier'.

- **Name:** Enter a name for the action.
- **Action Type:** Click the drop-down list to select an action type to either create or update contact records.
- **Connection:** The connection cannot be modified.
- **Object Mapping:** To see available fields for mapping, select an Eloqua Object and CX Sales Object and click **Fetch**. The unique identifiers for field mappings will be determined once the *Fetch* button is clicked. Top-level objects from your Oracle Sales instance are available in the **CX Sales Object** list.


 **Note:** The **Fetch** button is only available for Oracle CX Sales (formerly Oracle Sales Cloud) version R12 or higher. Older versions of Oracle CX Sales automatically load fields to be mapped.

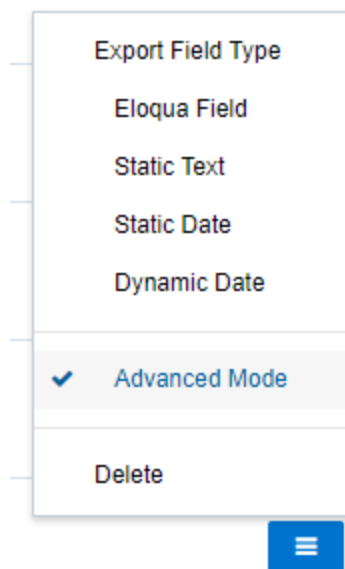
- **Update Oracle CX Sales Fields:** For each set of fields, specify the Oracle Sales field that will be mapped to Oracle Eloqua contacts. If you are updating contact records, the unique identifier is the first




mapping shown in this list. Add more mappings by clicking **Add Export Field Mapping**. Click the menu  to delete the mapping, or switch the field type to static text, static date, or dynamic date. You can also **allow blank fields to be sent** from Oracle Eloqua to Oracle Sales when using the *Update* action type. This option is only available for fields that are not required.

 **Warning:** Do not check the **Allow null fields to be sent** option when mapping to a Oracle Sales date field type. An error will occur.

- **Update Eloqua Fields:** For each set of fields, specify the Eloqua field that will be mapped to Oracle Sales contacts. If you are creating contact records, the unique identifier is the first mapping shown in this list. Add more mappings by clicking **Add Import Field Mapping**. Click the menu  to delete the mapping or switch a *From Oracle CX Sales* field type to static text, static date, or dynamic date.



- If you select **Advanced Mode** from the , you can enter multiple values to be sent to a single Oracle Sales field: **Eloqua Object**, **Date**, or **Dynamic Date**. You can also type static text into the box as needed. For contacts, you can use advanced mode to select contact and related account fields. For CDOs,

advanced mode allows you to select a CDO and its related contact and account fields.

App Configuration for Oracle Sales Cloud Integration

Edit Action

Business Phone ( C\_BusPhone ) Work Phone Number ( WorkPhoneNumber )

Zip or Postal Code ( C\_Zip\_Postal ) Postal Code ( PostalCode )

State or Province ( C\_State\_Prov ) State ( State )

Advanced Mode

Insert Object +

Insert Object

Eloqua Object

Date

Dynamic Date


Update Eloqua Fields

From Oracle Sales Cloud To Eloqua

- **Import Rules:** Specify data priority.
  - **Data Priority:** Select the priority of data to resolve conflicts on imports, such as Bulk API and CRM Integration. If leads and contacts are being imported to Eloqua contacts, there may be situations where a lead and contact with the same email address exist. Eloqua's data priority can be used to instruct Eloqua which source, lead or contact, is higher priority. If the contact import is set to have a higher priority than the lead import, the data imported from the contact will not be overwritten by the lower priority lead data. Higher priority sources are never overwritten by lower priority sources. Data priority needs to be specified as part of an action to ensure that higher priority sources are not overwritten by lower priority sources from other imports.

4. Click **Save**.

Your action is created and can be used in a [program or campaign](#).

Click the menu icon  to modify the action. You can edit, enable, disable, or view report data for the action. If an action is disabled, you can delete it. If you attempt to delete an action that is used on a canvas, a warning message will display, but it can be overwritten if desired.

| Action Name | From Source     | To Destination                       |
|-------------|-----------------|--------------------------------------|
| Example 1   | Eloqua: Contact | Sales Cloud: Sales Cloud Account SDO |
| Example 2   | Eloqua: Contact | Sales Cloud: Sales Cloud Account SDO |

View Reports

Edit

Disable

**Important:** If an action is being used within a program or campaign and the action is disabled, contacts will not be processed until the action is enabled. Once the action has been re-enabled, the app will process all contacts.

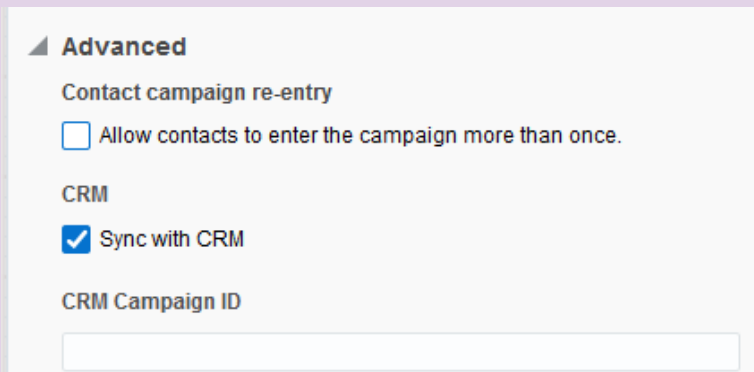
## Creating campaign actions

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

**Note:** This page outlines how to create campaign actions. Learn how to create [step actions](#) and [response actions](#).

Actions are data transactions between Oracle Eloqua and Oracle Sales. The *Campaign Actions* page is where you create actions to send data to Oracle Sales when campaigns change status. A status change is triggered by a campaign being created, deactivated, activated, scheduled, or completed. These actions are used for sending newly created campaigns, a campaign's activation status, deactivation status, completion status, and updates to existing campaigns (provided the Oracle Eloqua campaign has a Oracle Sales campaign ID).

**Tip:** To sync your campaign with Oracle Sales, check the **Sync with CRM** box in your campaign's advanced settings (open a campaign > [Options](#) > [Campaign Settings](#) > [Advanced](#)). The CRM Campaign ID field populates after Oracle Eloqua syncs with Oracle Sales for the first time. This field (CRM Id) is used as a unique identifier for campaign actions. [Map at least one more field](#) in addition to the CRM Id.



**Advanced**

Contact campaign re-entry

☐ Allow contacts to enter the campaign more than once.

CRM

☒ Sync with CRM

CRM Campaign ID

Campaigns in Oracle Sales may also be imported into Oracle Eloqua to allow for bi-directional updates between the two applications. When a campaign action is enabled, any campaign imports using different connections are automatically disabled.

## Prerequisites

- Configure the following campaign statuses in [Oracle CX Sales](#):

| Eloqua campaign status           | Campaign status codes sent to CX Sales |
|----------------------------------|--|
| Campaign created or de-activated | ORA_DRAFT                              |
| Campaign activated               | ORA_ACTIVE                             |
| Campaign scheduled               | ORA_SCHEDULED                          |
| Campaign completed               | ORA_COMPLETED                          |

- A [configured connection](#)

## To create a campaign action:

1. In the Oracle CX Sales Integration app, click **Campaigns**.
2. On the *Campaign Actions* tab, click **Add Action** and select a connection for which you want to create a new action.
3. Complete the action details:

App Configuration for Oracle Sales Cloud Integration

Create Action

Action Name: CBJA - Campaign

Connection: CBJA


Eloqua Object: Campaign

Sales Cloud Object: Campaign


Fetch


| From Eloqua                       | To Oracle Sales Cloud          | Blank Values      |
|-----------------------------------|--------------------------------|-------------------|
| CRM Id ( crmid )                  | Campaign ID ( CampaignId )     | Unique Identifier |
| Name ( name )                     | Name ( CampaignName )          |                   |
| Actual Cost ( actualCost )        | Actual Cost ( ActualCost )     |                   |
| Budgeted Cost ( budgetedCost )    | Budgeted Cost ( BudgetedCost ) |                   |
| Campaign End Date ( endDate )     | End Date ( EndDate )           |                   |
| Campaign Start Date ( startDate ) | Start Date ( StartDate )       |                   |


- **Name:** Enter a name for the action.
- **Object Mapping:** The Oracle Eloqua Object is set to *Campaign* and cannot be modified. To see available fields for mapping, select a top-level object from the **CX Sales Object** list and click **Fetch**. The unique identifiers for field mappings will be determined once the *Fetch* button is clicked.

 **Tip:** If enabled for your account, you can select Oracle CX Sales Campaign and Campaign Member objects. This feature is currently released under our Controlled Availability program. To request access to this feature, please log in to [My Oracle Support](https://support.oracle.com) (<https://support.oracle.com>) and create a service request.

- **Update Oracle CX Sales Fields:** The unique identifiers are automatically determined by the selected Eloqua and CX Sales objects in the *Object Mapping* section. The Oracle Eloqua unique identifier is *CRM Id*. You must add at least one additional mapping by clicking **Add Export Field Mapping**. For each set of the fields, specify the Oracle Sales field that will be mapped to Oracle Eloqua contacts.

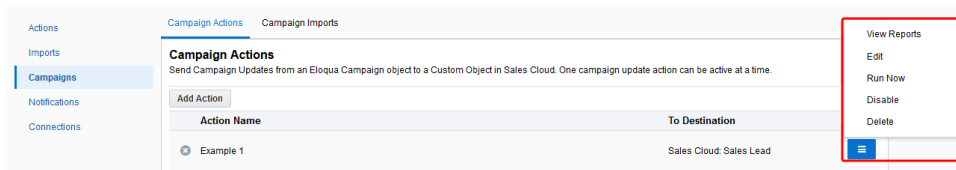
Click the menu  to delete the mapping, or switch the field type to static text, static date, or dynamic date. You can also **allow blank values to be sent** from Oracle Eloqua to Oracle Sales when using the *Update* action type. This option is only available for fields that are not required.

 **Warning:** Do not check the **Allow blank values to be sent** option when mapping to an Oracle Sales date field type. An error will occur.

- **Update Eloqua Fields:** The unique identifiers are automatically determined by the Eloqua and CX Sales objects specified in the *Object Mapping* section. The Oracle Eloqua unique identifier is *CRM Id*. You must add at least one additional mapping by clicking **Add Import Field Mapping**. For each set of fields, specify the Oracle Eloqua field that will be mapped to Oracle Sales contacts. Click the menu  to delete the mapping or switch a *From Oracle CX Sales* field type to static text, static date, or dynamic date.

4. Click **Save**.

Click the list to modify the action. You can edit, enable, disable, or view report data for the action. Only one campaign action can be enabled at a given time. Enabling a new action will disable an existing one. You can delete disabled actions. The action can be run on demand using the **Run Now** option, and selecting a specific campaign.



## Creating response actions

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

This page outlines how to create response actions. Learn how to create [step actions](#) and [campaign actions](#).

Actions are data transactions between Oracle Eloqua and Oracle CX Sales. The Response Actions page is where you create actions to send data to Oracle CX Sales based on response rules configured in your Oracle Eloqua instance. When a response occurs, Oracle Eloqua records the response based on priority and the response is sent to an Oracle CX Sales object.

**Important:** Avoid uploading contact activities that are older than 30 days. These contact activities will not be sent to the Oracle CX Sales integration app.

The below image shows an example of Oracle Eloqua response rules configured to use program and campaign canvas for campaign associations.

**Response Rules**

**Default Campaign Response Rules**

Default Campaign Response Rule Details

Eloqua entities that perform the response activities selected below will be associated with the campaign on which the activity was performed. A response higher in priority will overwrite a previous response.

|   | Priority | Response Activity  | Oracle Sales Cloud Member Campaign Status | Oracle Sales Cloud Responded | Oracle Sales Cloud Default |
|---|----------|--|---|------------------------------|----------------------------|
| X | 1        | External Activity: Tradeshow Visited Booth               | (none)                                    | ✓                            | ○                          |
| X | 2        | External Activity: Shopping Cart Check-out Made Purchase | (none)                                    | ✓                            | ○                          |
| X | 3        | Response - Form Submit                                   | (none)                                    | ✓                            | ○                          |
| X | 4        | External Activity: Tradeshow Viewed a Demo               | (none)                                    | ✓                            | ○                          |
| X | 5        | External Activity: Response Test Responded               | (none)                                    | ✓                            | ○                          |

+ Add Response Activity

Oracle Sales Cloud Integration

Step 1: Define which Oracle Sales Cloud entities are created or updated.

Set up integration rules if you wish to perform integration actions like creating or updating leads and contacts in Oracle Sales Cloud. The integration rules will run when an Eloqua entity (contact, prospect or company) has performed a response on a campaign.

| Rule Name                               | Entity | Entity Filter | Integration Event |
|---|--------|---------------|-------------------|
| There are no integration rules defined. |        |               |                   |

+ Add Rule

Step 2: Associating a campaign (and status) to a Oracle Sales Cloud entity.

☐ Use Program Builder for Campaign Associations

☒ Use Program and Campaign Canvas for Campaign Associations

App: Oracle Sales Cloud Integration

Service: Oracle Sales Cloud Campaign Response Integration

Instance: Oracle Sales Cloud Campaign Response Integration (Program - CRM - Campaign Response)

☐ Use Integration Rules for Campaign Associations


## Prerequisites

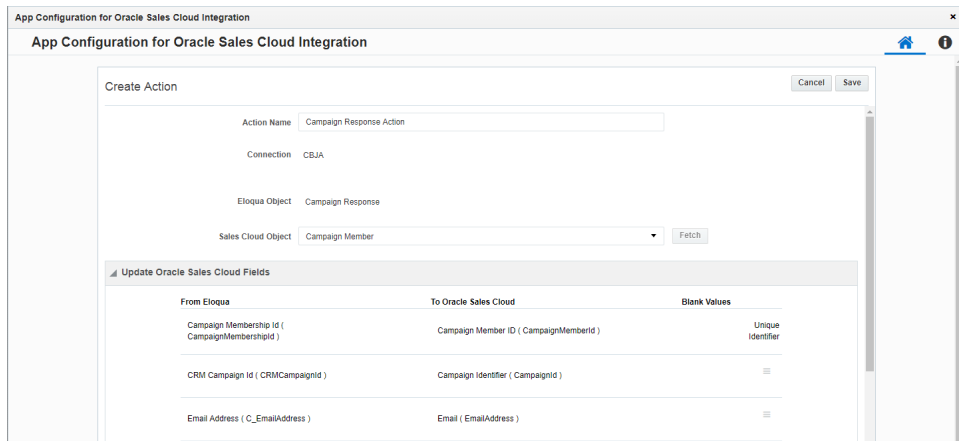
- A [configured connection](#)
- A response rule configured in your Oracle Eloqua instance



## To create a response action:


1. In the Oracle CX Sales Integration app, click **Campaigns**.
2. Click the **Response Actions** tab.



3. Click **Add Action** and select a connection for which you want to create a new action.
4. Complete the action details:
  - **Action Name:** Enter a name for the action.
  - **Connection:** The connection cannot be modified.
  - **Object Mapping:** The **Eloqua Object** is set to *Campaign Response* and cannot be modified. Select a top-level object from the **CX Sales Object** list and then click **Fetch**. The default field mapping is displayed.
  - **Update Oracle CX Sales Fields:** The Oracle Eloqua unique identifier is *Campaign Membership Id*. The Oracle CX Sales unique identifier is determined by the CX Sales object selected in *Object Mapping*. Add more mappings by clicking **Add Export Field Mapping**. For each set of fields, specify the Oracle CX Sales field that will be mapped to Oracle Eloqua contacts. Click the menu  to delete the mapping, or switch the field type to static text, static date, or dynamic date. You can also **allow blank values to be sent** from Oracle Eloqua to Oracle CX Sales when using the *Update* action type. This option is only available for fields that are not required.

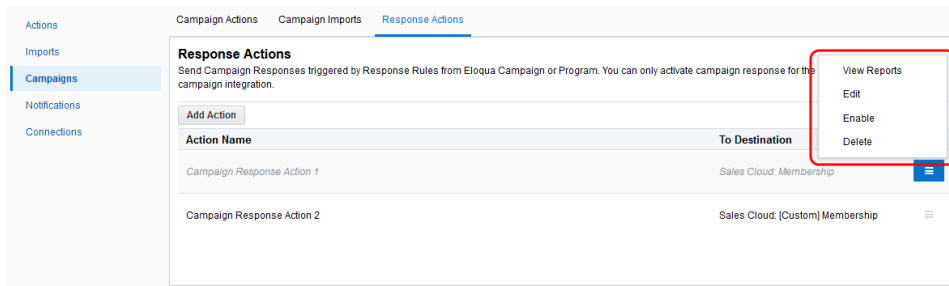


| From Eloqua                                     | To Oracle Sales Cloud                   | Blank Values  |
|---|---|---|
| Campaign Membership Id ( CampaignMembershipId ) | Campaign Member ID ( CampaignMemberId ) | Unique Identifier   |
| CRM Campaign Id ( CRMCampaignId )               | Campaign Identifier ( CampaignId )      |  |
| Email Address ( C_EmailAddress )                | Email ( EmailAddress )                  |  |

 **Warning:** Do not select the **Allow blank values to be sent** option when mapping to an Oracle CX Sales date field type. An error will occur.

5. Click **Save**.

Click the list to modify the action. You can edit, enable or disable, and view report data for the action. Only one campaign action can be enabled at a time. Enabling a new action will disable an existing one. You must disable an action before you can delete it.



## Creating imports

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

**Note:** This page outlines how to manually create imports in the Oracle CX Sales Integration app. You can also [create imports automatically](#).

Imports are data transactions from Oracle CX Sales into Oracle Eloqua. The Oracle CX Sales Integration app executes imports every 15 minutes. After an import has been scheduled, an import schedule does not display because imports are executed so often. If an import needs to be performed off schedule, it can be executed on demand. An import may fail due to various reasons such as connection or data related issues. Some of these

issues will cause an entire import to fail. You can setup email [notifications](#) to alert you when import errors occur.

The following imports are recommended:

- Accounts
- Contacts
- Leads
- Opportunities

When viewing your imports, disabled imports are dimmed to indicate their status. There are several indicators in your list of imports to help you quickly get an overview of their status:

The screenshot shows the 'Imports' section of the Oracle interface. A table lists various imports with columns for 'Import', 'From Source', 'To Destination', and 'Last Import'. Callouts point to specific indicators: a green checkmark for 'The last import was successful', a greyed-out row for 'The import is disabled', a red 'X' for 'The last import was unsuccessful', and a yellow warning triangle for 'The last import has warnings'. A red box highlights the 'Import' column header with the text 'The import is currently running'.

| Import                         | From Source                            | To Destination    | Last Import           |
|--------------------------------|--|-------------------|-----------------------|
| CAMX - Activity Import         | Sales Cloud: Sales Lead                | Eloqua: Contact   | Today At 7:57 AM      |
| CAMX - Contact Import          | Sales Cloud: Contact                   | Eloqua: Contact   | 07/19/2017            |
| CAMX - Imports                 | Sales Cloud: Sales Cloud Household SDO | Eloqua: SX Global | 10/19/2017            |
| CAMX - Account Import          | Sales Cloud: Sales Cloud Account SDO   | Eloqua: Contact   | Today At 10:12 AM     |
| CAMX - 29072018 -001 - CONTACT | Sales Cloud: Contact                   | Eloqua: Contact   | Yesterday At 11:59 PM |

## Prerequisites

- A [configured connection](#)
- Create any [shared contact lists](#) if needed for [post processing](#) the import.

**Note:** Campaign, opportunity, and user imports have additional prerequisites.

## To create an import:

1. In the Oracle CX Sales Integration app, click **Imports**.
2. Click **Add Import** and select the connection for which you want to create a new import.
3. Complete the import details:
  - **Name:** Enter a name for the import.
  - **Deleted or Merged Records:** Select this option if you want to recognize deleted or merged (that is, non-survivor) Oracle Sales records in Oracle Eloqua. If you select this option, the **CX Sales Object** and **Eloqua Object** lists only display the **Account** and **Contact** options.

Create Import

Import Name CBVE - New Import

Connection CBVE

☐ Deleted and Merged Records ☒ Exclude ☐ Import Only Deleted Records ☐ Import Only Merged Records

CX Sales Object Sales Lead Fetch

Filter Test

Eloqua Object Contact

- **CX Sales Object:** Select the top-level objects from your Oracle CX Sales instance that you want to import and click **Fetch**. Oracle CX Sales is queried and additional options are displayed and unique identifiers for field mappings are available on the *Create Import* page.
- **Filter:** (Optional) Enter a filter to use on the source field when querying contacts to import and then click **Test** to test the filter.  
See example filters

To learn more about filters, see the [Oracle CX Sales \(formerly Oracle Sales Cloud\) REST API guide](#).

| Filter Syntax  | Description                       |
|--|-----------------------------------|
| <i>FirstName=&lt;value&gt;</i><br><b>Example:</b> FirstName='Thomas' | Filters on a record's first name. |
| <i>LastName=&lt;value&gt;</i><br><b>Example:</b> LastName='Lambert'  | Filters on a record's last name.  |

| Filter Syntax   | Description   |
|---|---|
| <i>Title=&lt;value&gt;</i><br><b>Example:</b> Title='Marketer'  | Filters on a record's title.  |
| <i>PartyStatus!=&lt;value&gt;</i><br><b>Example:</b> PartyStatus!='A'   | Filters on a record's party status that is not active. A value of "A" means the status is active. |
| <i>LastName LIKE &lt;value&gt;%</i><br><b>Example:</b> LastName LIKE 'Thomas%'  | Filters on a record's last name to compare it with a value using a wildcard operator.             |
| <i>PrimaryContactEmailAddress LIKE<br/>%@%</i><br><b>Example:</b> PrimaryContactEmailAddress<br>LIKE '%@%'                                | Filters on a record's email address to compare it with a value using a wildcard operator.         |
| <i>LastName LIKE &lt;value&gt;% AND<br/>FirstName=&lt;value&gt;</i><br><b>Example:</b> LastName LIKE 'Lambert%'<br>AND FirstName='Thomas' | Filters on a record's first name and last name.   |

This is an example of test filter results:

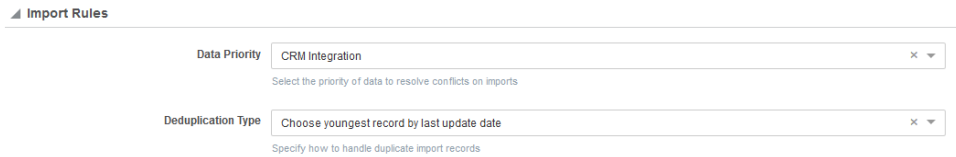
Displaying 10 of 1,186 [View full source](#)

|                             |                                     |
|-----------------------------|-------------------------------------|
| Record 1                    | <a href="#">Copied to clipboard</a> |
| PartyNumber                 | 4003                                |
| AccountName                 | null                                |
| FirstName                   | Jason                               |
| LastName                    | Test                                |
| Title                       | null                                |
| EmailAddress                | jason.test@thelambert-test.com      |
| PrimaryAddress.AddressLine1 | 123 real street dr                  |
| PrimaryAddress.AddressLine2 | null                                |
| PrimaryAddress.AddressLine3 | null                                |
| PrimaryAddress.City         | San Francisco                       |
| PrimaryAddress.Country      | US                                  |
| PrimaryAddress.PostalCode   | 94110                               |
| PrimaryAddress.State        | CA                                  |
| Record 2                    | <a href="#">Copy to clipboard</a>   |
| Record 3                    | <a href="#">Copy to clipboard</a>   |
| Record 4                    | <a href="#">Copy to clipboard</a>   |

- **Eloqua Object:** Select an Oracle Eloqua object and Oracle CX Sales object .
- **Update Eloqua Fields:** Map your unique identifier to specify the object in Oracle Eloqua to update or create. For each set of fields, select the Oracle Eloqua field that will be mapped to Oracle CX Sales

contacts. Add more mappings by clicking **Add Import Field Mapping**. Click the menu  to delete the mapping or switch the field type to static text, static date, or dynamic date.

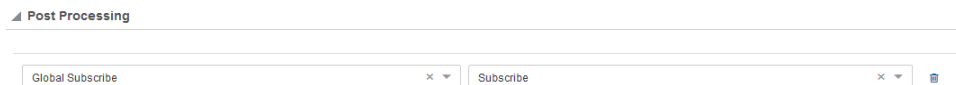
- **Import Rules:** Specify data priority and deduplication rules.



The 'Import Rules' configuration interface shows two settings:

- Data Priority:** Set to 'CRM Integration'. Below the dropdown is the text: 'Select the priority of data to resolve conflicts on imports'.
- Deduplication Type:** Set to 'Choose youngest record by last update date'. Below the dropdown is the text: 'Specify how to handle duplicate import records'.

- **Data Priority:** Select the priority of data to resolve conflicts on imports, such as Bulk API and CRM Integration. If leads and contacts are being imported to Oracle Eloqua contacts, there may be situations where a lead and contact with the same email address exist. Oracle Eloqua's data priority can be used to instruct Oracle Eloqua which source, lead or contact, is higher priority. If the contact import is set to have a higher priority than the lead import, the data imported from the contact will not be overwritten by the lower priority lead data. Higher priority sources are never overwritten by lower priority sources.
- **Deduplication type:** Specify how to handle duplicate import records. For example, you can choose the most recently updated record or import all records. Deduplication runs per import execution. If multiple records with the same match field value are imported in a single execution, those records will be deduplicated based on the setting specified here.
- **Post Processing:** Specify an action to be performed on a contact after import. You can specify a maximum of ten post processing items, including the following actions and any OSC Feeder elements that you [add to a canvas](#).



The 'Post Processing' configuration interface shows a list of actions:








- Global Subscribe
- Subscribe

- **Add to Shared List / Remove from Shared List:** Specify whether contacts should be added to or removed from an existing [shared contact list](#). As contacts are processed, Oracle Eloqua adds them to the shared list that you specify.
- **Subscribe to Email Group / Unsubscribe from Email Group:** Specify whether contacts should be subscribed to or unsubscribed from an email group.
- **Global Subscribe:** Specify whether to apply the global **Subscribe** or **Unsubscribe** actions to the contacts.
- **SMS Opt-in Phone Number / SMS Opt-out Phone Number:** Specify whether after the import, phone number(s) should be opted in or opted out from receiving SMS messages.
- **Add Action:** Click to add another action to be performed after the import.

4. **Program Usage:** Once you save the import and [associate it with a program](#) on the campaign or program canvas to perform any specified post processing actions, this section will display the associated program dependencies and provide a link to display the canvas.
5. Click **Save**.

## Viewing and editing imports

### To view or edit an import:


1. In Eloqua, click the *cloud menu*  icon on the right hand side to expand the *Apps* list and select **Oracle CX Sales Integration**. The *App Configuration for Oracle CX Sales Integration* page is displayed.
2. Click **Imports**. The list of imports provides several indicators and each import has a menu  with several options.
  - Colored icons indicate that the import is enabled.
  - Disabled imports are dimmed to indicate their status.
  -  : Indicates that the import is currently running. When enabled, imports run automatically every 15 minutes.
  -  : Indicates that the last import was successful.
  -  : Indicates that the last import was unsuccessful.
  -  : Indicates that the last import failed with warnings.
  -  : Indicates that the import is associated with a program that is running on a canvas, such as the [OSC Feeder action](#). If you attempt to disable or delete an import with an associated program that is running, you will receive a warning message.

**Imports**  
Connection imports are automatically started 15 minutes after the previous import has finished.

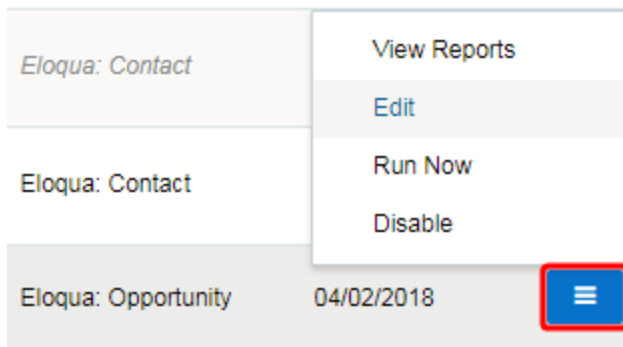
**Callouts:**

- The import is currently running:** Points to the 'CAMX - Activity import' row.
- The last import was successful:** Points to the 'CAMX - Contact Import' row.
- The import is disabled:** Points to the 'CAMX - Imports' row.
- The last import was unsuccessful:** Points to the 'CAMX - Account' row.
- The last import has warnings:** Points to the 'CAMX - 26072018 -001 - CONTACT' row.
- The import is associated with an item on the program canvas:** Points to the 'CAMX - 26072018 -001 - CONTACT' row.

| Import                         | From Source                            | To Destination    | Last Import           |
|--------------------------------|--|-------------------|-----------------------|
| CAMX - Activity import         | Sales Cloud: Sales Lead                | Eloqua: Contact   | Today At 7:57 AM      |
| CAMX - Contact Import          | Sales Cloud: Contact                   | Eloqua: Contact   | 07/19/2017            |
| CAMX - Imports                 | Sales Cloud: Sales Cloud Household SDO | Eloqua: SX Global | 10/19/2017            |
| CAMX - Account                 | Sales Cloud: Sales Cloud account SDO   | Eloqua: Contact   | Today At 10:12 AM     |
| CAMX - 26072018 -001 - CONTACT | Sales Cloud: Contact                   | Eloqua: Contact   | Yesterday At 11:59 PM |

3. Click the import's menu  to select options:


- **View Reports:** Display activity for the import by date.
- **Edit:** Modify some of the import's values or view its details, including the *Program Usage* section.
- **Enable or Disable:** If you enable the import, the **Run Now** option is displayed, which allows you to run the import on demand.
- **Delete:** Once an import is disabled, the **Delete** option is displayed.



4. Click **Edit** to view the import's details, such as mapping, import rules, and so on. You can edit any attributes except for:

- Connection
- CX Sales Object
- Eloqua Object
- Program Usage items

5. To view the program associated with the import, expand the import's *Program Usage* section

and click the go to icon  to display the associated [campaign or program canvas](#).



## Edit Import

| ► Import Rules   |                       |                |   |
|--|-----------------------|----------------|---|
| ► Post Processing  |                       |                |   |
| ▲ Program Usage  |                       |                |   |
| This import is currently being used in the following Programs. |                       |                |   |
| Step Name  | Program Name          | Program Status |   |
| OSC Feeder   | OSC Feeder Campaign   | 🚧 Draft        | 📄 |
| OSC Feeder   | OSC-18C-REG-001       | 🟢 Running      | 📄 |
| OSC Feeder   | OSC Feeder Campaign 2 | 🚧 Draft        | 📄 |

## Scheduling an import

Once you create an import, it is scheduled to run automatically every 15 minutes.

You can also run the import on demand using the **Run Now** option. When **Run Now** is used with a date and time filter, the current date and time is pre-populated. The date and time can be changed to a time in the past as desired. If no date is selected, the import retrieves all matching records from Oracle CX Sales and imports them to Oracle Eloqua. Selecting a specific date allows you to recover missed records and perform partial imports if field mapping changes.

**Note:** The *Last Import* date indicates the last successful import.

Clicking **Disable** temporarily stops the imports. If the import is currently running and has not yet started syncing to Oracle Eloqua (if it is in either the *Retrieve from CX Sales* or *Deduplicating* steps), the import is canceled and disabled. If the import has begun importing to Oracle Eloqua, the import will finish, then be disabled. Click **Enable** to schedule your import to run every 15 minutes.

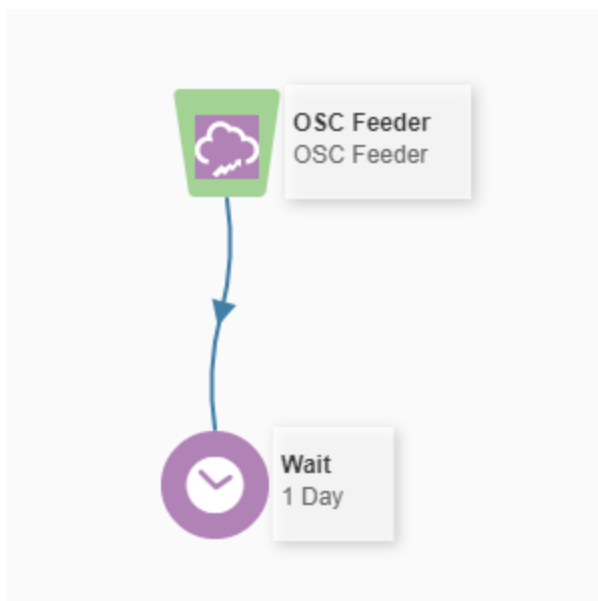
You can also [add imports to a canvas](#) and schedule them via the Wait element.

## Adding imports to a canvas

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.


Using Oracle Eloqua's OSC Feeder service, you can specify an Oracle CX Sales Integration app import and schedule the import's **post processing** actions. When the feeder runs, the post processing actions update the associated import.

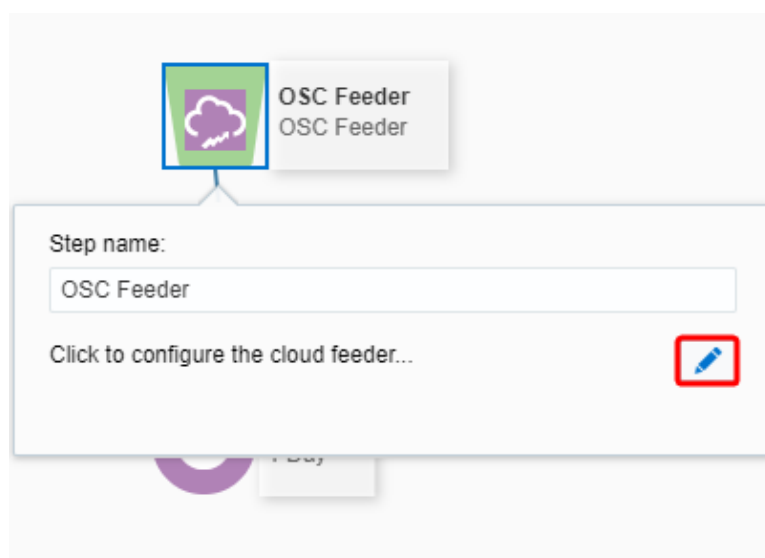
You can use this service on the program canvas. The following example shows a program canvas where the *Wait* element is configured to run the OSC Feeder step at specific intervals. You can also use the OSC Feeder element with other audience steps on the program canvas to **update or create contact records within a program**.



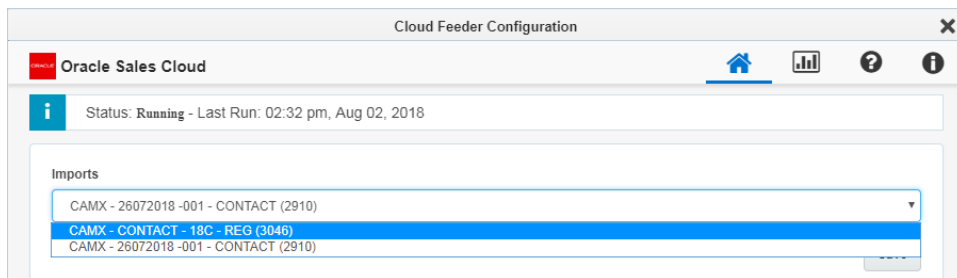
**Note:** Adding the OSC Feeder element to a canvas counts towards the limit of ten post processing actions.

### To add the OSC Feeder to a program:

1. Open an existing program or create a new one.
2. From the *Data Sources* section, drag the OSC Feeder element onto the canvas.
3. Configure the [Wait element](#) to run at specific intervals and connect it to the OSC Feeder step.
4. Click **Save**. The canvas must be saved before you can configure the OSC Feeder step.
5. Double-click the element to open its configure screen. The window that opens enables you to do the following:
  - Change the name of the step to something more meaningful.
  - Click the edit icon  to configure the element.



6. Click the edit icon to configure the action. The configuration dialog provides the option to select an import.



**Tip:** If you are using Firefox and the configuration screen does not load properly, clear the browser cache.

7. From the **Imports** list, select the import you want to associate with the OSC Feeder. This creates a link between the canvas and the import.
8. Click **Save** to save the configuration and then click **Save** to save any changes you made to the program.

When imports flow through the OSC Feeder element, its **post processing** actions are run at the intervals scheduled by the Wait element.

To view a report about the information the OSC Feeder has processed, its status, errors, and other details, view its **Daily Feeder Run report**.

# Creating campaign imports

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

You can create and schedule campaign imports in the Oracle CX Sales Integration app. The Oracle CX Sales Integration app executes imports every 15 minutes. After an import has been scheduled, an import schedule does not display because imports are executed so often. If an import needs to be performed off schedule, it can be executed on demand. An import may fail due to various reasons such as connection or data related issues. Some of these issues will cause an entire import to fail. You can setup email [notifications](#) to alert you when import errors occur.

When a **campaign action** is enabled, any campaign imports using different connections are automatically disabled.

## Prerequisites

- A **configured connection**


## To create a campaign import:

1. In the Oracle CX Sales, click **Campaigns**.
2. Click the **Campaign Imports** tab.
3. Click **Add Import** and select a connection for which you want to create a new import.

#### 4. Complete the import details:

The screenshot shows the 'App Configuration for Oracle Sales Cloud Integration' window. The 'Create Import' section has the following fields: 'Import Name' (CBJA - New Campaign Import), 'Connection' (CBJA), 'Sales Cloud Object' (Campaign), and 'Eloqua Object' (Campaign). There is a 'Filter' input field and a 'Test' button. The 'Update Eloqua Fields' section shows a mapping from 'Campaign ID ( Campaignid )' to 'CRM id ( crmid )' as the 'Unique Identifier', and 'Name ( CampaignName )' to 'Name ( name )'.

- **Name:** Enter a name for the import.
- **Object Mapping:** The **Eloqua Object** is set to *Campaign* and cannot be modified. To see available fields for mapping, select a top-level object from the **CX Sales Object** list and click **Fetch**. The unique identifiers for field mappings will be determined once the *Fetch* button is clicked.

 **Note:** The **Fetch** button is only available for Oracle CX Sales (formerly Oracle Sales Cloud) version R12 or higher. Older versions of Oracle CX Sales automatically load fields to be mapped.


- **Filter:** Enter a filter to use on the CRM source field when querying contacts to import. Click **Test** to test the filter.

See example filters

To learn more about filters, see to the [Oracle CX Sales REST API Guide](#).

| Filter Syntax  | Description                       |
|--|-----------------------------------|
| <i>FirstName=&lt;value&gt;</i><br><b>Example:</b> FirstName='Thomas' | Filters on a record's first name. |
| <i>LastName=&lt;value&gt;</i><br><b>Example:</b> LastName='Lambert'  | Filters on a record's last name.  |
| <i>Title=&lt;value&gt;</i>   | Filters on a record's title.      |

| Filter Syntax   | Description   |
|---|---|
| <b>Example:</b> Title='Marketer'                                |   |
| <i>PartyStatus!=&lt;value&gt;</i>                               | Filters on a record's party status that is not active. A value of "A" means the status is active. |
| <b>Example:</b> PartyStatus!='A'                                |   |
| <i>LastName LIKE &lt;value&gt;%</i>                             | Filters on a record's last name to compare it with a value using a wildcard operator.             |
| <b>Example:</b> LastName LIKE 'Thomas%'                         |   |
| <i>PrimaryContactEmailAddress LIKE<br/>%@%</i>                  | Filters on a record's email address to compare it with a value using a wildcard operator.         |
| <b>Example:</b><br>PrimaryContactEmailAddress LIKE<br>'%@%'     |   |
| <i>LastName LIKE<br/>&lt;value&gt;%;FirstName=&lt;value&gt;</i> | Filters on a record's first name and last name.   |
| <b>Example:</b> LastName LIKE<br>'Lambert%';FirstName='Thomas'  |   |

- **Update Eloqua Fields:** The unique identifiers are automatically determined by the **Eloqua Object** and **CX Sales Object** values in the *Object Mapping* section. For each set of fields, select the Oracle Eloqua field that will be mapped to Oracle CX Sales contacts. Add more mappings by clicking **Add Import Field Mapping**. Click the menu  to delete the mapping or switch the field type to static text, static date, or dynamic date.
- **Import Rules:** Specify deduplication rules.
  - **Deduplication type:** Specify how to handle duplicate import records. For example, you can chose the most recently updated record or import all records. Deduplication runs per import execution. If multiple records with the same match field value are imported in a single execution, those records will be deduplicated based on the setting specified here.
- **Post Processing:** Post processing rules are not available for campaign imports.

5. Click **Save**.

Click the import's menu to modify the import. You can edit, enable (to schedule your import) and disable, or view report data for an import. If an import is disabled, you can delete it.

|                 |                          |                  |                      |  |
|-----------------|--------------------------|------------------|----------------------|--|
| Example Import1 | Sales Cloud: Lead        | Eloqua: Campaign | 09/23/2016           | View Reports<br>Edit<br>Enable<br>Delete |
| Example Import2 | Sales Cloud: Opportunity | Eloqua: Campaign | 08/16/2016           |  |
| Example Import3 | Sales Cloud: Contact     | Eloqua: Campaign | Today At 3:32 PM     |  |
| Example Import4 | Sales Cloud: Account     | Eloqua: Campaign | Yesterday At 9:34 PM |  |

Once your import is scheduled, it automatically runs every 15 minutes. The import can be run on demand using the **Run Now** option. When **Run Now** is used with a date and time filter, the current date and time is pre-populated. The date and time can be changed to a time in the past as desired. If no date is selected, the import retrieves all matching records from Oracle CX Sales and imports them to Oracle Eloqua. Selecting a specific date allows you to recover missed records and perform partial imports if field mapping changes.

**Note:** The *Last Import* date indicates the last successful import.

Clicking **Disable** temporarily stops the imports. If the import is currently running and has not yet started syncing to Oracle Eloqua (if it is in either the *Retrieve from CX Sales* or *Deduplicating* steps), the import is canceled and disabled. If the import has begun importing to Oracle Eloqua, the import will finish, then be disabled. Click **Enable** to enable your import.

Disabled imports are dimmed to indicate their status. There are several indicators in your list of imports to help you quickly get an overview of their status:



|               |                  |                  |
|---------------|------------------|------------------|
| Actions       | Campaign Actions | Campaign Imports |
| Imports       |                  |                  |
| Campaigns     |                  |                  |
| Notifications |                  |                  |
| Connections   |                  |                  |

| Campaign Imports  |   |                  |                   |  |
|---|---|------------------|-------------------|--|
| Update Eloqua campaigns from CRM. Campaign imports are automatically started 15 minutes after the previous import has finished. |   |                  |                   |  |
| Add Import  |   |                  |                   |  |
| Import  | From Source                                     | To Destination   | Last Import       |  |
| Campaign Import1  | Sales Cloud: [Custom] Eloqua Marketing Campaign | Eloqua: Campaign | Today At 7:57 AM  |  |
| Campaign Import2  | Sales Cloud: [Custom] Eloqua Marketing Campaign | Eloqua: Campaign | Today At 10:12 AM |  |
| Campaign Import3  | Sales Cloud: [Custom] Eloqua Marketing Campaign | Eloqua: Campaign | 10/19/2017        |  |
| Campaign Import4  | Sales Cloud: [Custom] Eloqua Marketing Campaign | Eloqua: Campaign | Today At 10:12 AM |  |

## Creating opportunity imports

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

**Note:** This page outlines how to manually create opportunity imports in the Oracle CX Sales Integration app. You can also [create imports automatically](#).

An opportunity is a potential sales deal created and maintained by the sales team. You can import Oracle CX Sales opportunities into Oracle Eloqua to enable marketers to report on opportunities and measure campaign effectiveness. Data imported by the Oracle CX Sales Integration app is required to populate [Insight's closed-loop reporting data](#) for Oracle CX Sales opportunities.

Opportunity imports have dependencies: they are typically linked to contacts directly or to contacts via accounts. The *Imports* list will display the opportunity import as a child of its parent object.

|                                 |   |                     |                  |   |
|---------------------------------|---|---------------------|------------------|---|
| ✓ New Connection - New Import   | Sales Cloud: Sales Cloud<br>Contact SDO | Eloqua: Contact     | Today At 5:48 PM | ⋮ |
| ✓ ↶ New Connection - New Import | Sales Cloud: Opportunity                | Eloqua: Opportunity | Today At 5:33 PM | ⋮ |

You must specify at least one parent dependency so that the linked Contact object is created in Oracle Eloqua *before* the opportunity is imported. You can specify up to two parent dependencies (typically a Contact and an Account import). To avoid opportunity import errors, parent imports should be successfully run before the opportunity import is run.

## Prerequisites

- A [configured connection](#)
- Identify the correct names of the object and associated fields that you want to import from Oracle CX Sales and use for linking to an Oracle Eloqua object. The linking field is the Oracle CX Sales Contact ID or Account ID.
- A successful [contact import](#) for the connection, which must be enabled before the opportunity import can be run
- If you will select **Contacts via Account**, you will also need a successful account import for the connection and link the contact import to the account. The linked account and contact imports must both be enabled before the opportunity import can be run.

## To create an opportunity import:

1. In the Oracle CX Sales Integration app, click **Imports**.
2. Click **Add Import** and select a connection for which you want to create a new import. The *Create Import* page displays a default name for the import.
3. Enter a unique name for the opportunity import.

4. From the **CX Sales Object** list, select **Opportunity**.
5. From the **Eloqua Object** list select **Opportunity**. The *Opportunity Linking* section is displayed.
6. Click **Fetch** to populate available fields in the *Opportunity Linking* section's lists.
7. From the **Eloqua Entity to link Opportunity records to** list, select one of the following:
  - **Contact**: Indicates that the Oracle Eloqua opportunity import should be linked to the contact import (one parent dependency).
  - **Contact via Account**: Indicates that the Oracle Eloqua opportunity import should be linked to the contact import, which is linked with its account import (two parent dependencies). The contact import should first be linked to the account import before the opportunity import is enabled.

**App Configuration for Oracle Sales Cloud Integration**

Create Import Cancel Save

Import Name: Marketing opportunity - New Connection

Connection: New Connection

Sales Cloud Object: Opportunity Fetch

Filter: Test

Eloqua Object: Opportunity

**Opportunity Linking: Specify the fields to be used for linking Opportunity records to Contacts or Accounts**

Eloqua Entity to link Opportunity records to: Contacts via Account

Eloqua Entity field used for linking: Contact


Sales Cloud Object field used for linking: Account (TargetPar


**Dependencies**


| Parent Import                      | Parent Source Object | Parent Destination Object |
|------------------------------------|----------------------|---------------------------|
| <span>Add Import Dependency</span> |                      |                           |

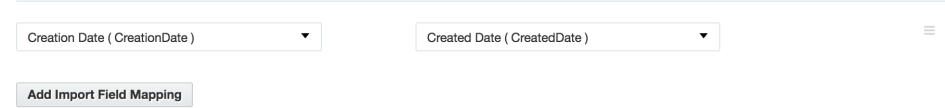
8. From the **CX Sales Object field used for linking** list, select one of the following Oracle CX Sales fields:
  - **Primary Contact (KeyContactId)**: If you are linking directly to contacts, select this field to link to the corresponding Oracle Eloqua contact field.
  - **Account (TargetPartyId)**: If you are linking to contacts via accounts, select this field to link to the corresponding Oracle Eloqua contact field.
9. From the **Eloqua Entity field used for linking** list, select one of the following Oracle Eloqua fields:


- **SC\_Contact\_PartyId:** If you are linking directly to contacts, select this field to link to the corresponding Oracle CX Sales contact field.
- **SC\_Account\_PartyId:** If you are linking to contacts via accounts, select this field to link to the corresponding Oracle CX Sales contact field.

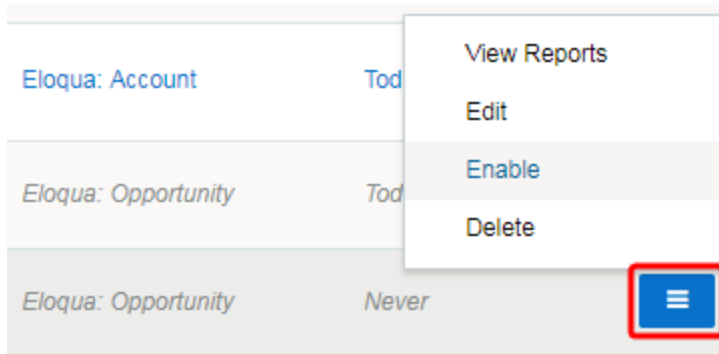
 **Note:** You can select different fields to use for linking if you have specific business needs.

10. In the *Dependencies* section, click **Add Import Dependency**.
11. In the *Parent Import* column, select the name of the parent object to establish the opportunity import's dependency on an existing import, such as the name of an existing contact import. If you have two parent dependencies, click **Add Import Dependency** again and add the second dependency. You can establish the dependency while the parent import is disabled. However, the parent import must be enabled before you can enable its opportunities import.
12. In the *Update Eloqua Fields* section, adjust the default field mappings if necessary. For each set of fields, select the Oracle Eloqua field that will be mapped to Oracle CX Sales contacts. You can add more mappings by clicking **Add Import Field Mapping**. Click the menu  to delete a mapping or switch its field type if needed.

 **Important:** To address a known issue, you must map Oracle CX Sales's **Creation Date (CreationDate)** field to Oracle Eloqua's **Created Date (CreatedDate)** field. If you do not add it, an error will be displayed when you attempt to enable the import. In a future release, this field will be mapped by default.



13. In the [Import Rules section](#), adjust the default data priority or deduplication rules if needed.
14. Click **Save**. The *Imports* page displays the new opportunity import in the list.
15. If one or more of the opportunity import's parent objects are disabled, click the corresponding menu  and select **Enable** to run the parent import.
16. Click the menu for the new opportunity import and select **Enable** to run it.



Once your import is scheduled, it automatically runs every 15 minutes and is based on the parent import timing and success. If there is a failure on any parent, the child import will not execute.

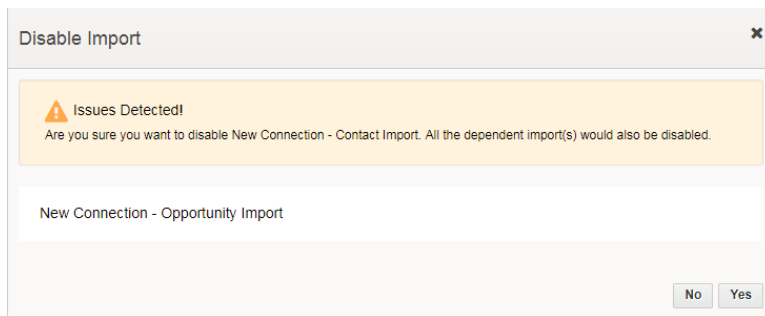
You can determine whether an opportunity import was successful or partially succeeded by selecting **View Report** from the menu. Some links for the object dependencies should be created for an initial successful opportunity import.

**Note:** The opportunity import report may indicate 0 links after the initial successful import if no additional records were created.

Clicking **Disable** temporarily stops the imports. If the import is currently running and has not yet started syncing to Oracle Eloqua (if it is in either the *Retrieve from CX Sales* or

*Deduplicating* steps), the import is canceled and disabled. If the import has begun importing to Oracle Eloqua, the import will finish, then be disabled.

If you click **Disable** or **Delete** for an import that has dependencies, a message will indicate those dependencies.



Before clicking **Yes** to disable or delete the import, determine whether you also want to disable the linked imports (or delink the imports if you are deleting a parent import).

## Creating user imports

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

**Note:** This page outlines how to manually create and schedule user imports in the Oracle CX Sales Integration app. You can also [create imports automatically](#).

Oracle CX Sales users can be imported into Oracle Eloqua. By importing, users can be created and updated, they can be placed in a folder, and added to or removed from a security group.

The Oracle CX Sales Integration app executes imports every 15 minutes. After an import has been scheduled, an import schedule does not display because imports are executed so often. If an import needs to be performed off schedule, it can be executed on demand. An import may fail due to various reasons such as connection or data related issues. Some of these issues will cause an entire import to fail. You can setup email [notifications](#) to alert you when import errors occur.

Disabled imports are dimmed to indicate their status. There are several indicators in your list of imports to help you quickly get an overview of their status:

The screenshot shows the 'Imports' section of the Oracle CX Sales interface. A sidebar on the left contains links for 'Actions', 'Imports', 'Campaigns', 'Notifications', and 'Connections'. The main area displays a table of imports with columns: 'Import', 'From Source', 'To Destination', and 'Last Import'. Callouts point to various status indicators: a blue progress bar for 'The import is currently running.', a green checkmark for 'The last import was successful.', a greyed-out row for 'The import is disabled.', a red 'X' icon for 'The last import was unsuccessful.', and a yellow warning triangle for 'The last import has warnings.'.


| Import                         | From Source                            | To Destination    | Last Import           |
|--------------------------------|--|-------------------|-----------------------|
| CAMX - Activity Import         | Sales Cloud: Sales Lead                | Eloqua: Contact   | Today At 7:57 AM      |
| CAMX - Contact Import          | Sales Cloud: Contact                   | Eloqua: Contact   | 07/19/2017            |
| CAMX - Imports                 | Sales Cloud: Sales Cloud Household SDO | Eloqua: SX Global | 10/19/2017            |
| CAMX - Account                 | Sales Cloud: Sales Cloud account SDO   | Eloqua: Contact   | Today At 10:12 AM     |
| CAMX - 26072018 -001 - CONTACT | Sales Cloud: Contact                   | Eloqua: Contact   | Yesterday At 11:59 PM |

## Prerequisites


- [A configured connection](#)
- For *User (BI report)* object, you will need to include the report path of a BI user report in Oracle CX Sales

**To create a user import:**


1. In the Oracle CX Sales Integration app, click **Imports**.
2. Click **Add Import** and select a connection for which you want to create a new import.
3. Complete the import details:
  - **Name:** Enter a name for the import.
  - **Connection:** The connection cannot be modified.
  - **CX Sales Object:** Select **User** or **User (BI report)**.


 **Note:** User (BI report) does not display for all accounts.

- **Report Path:** If you selected *User (BI report)*, enter in the path to the user report in Oracle CX Sales.

 **Note:** For the *User (BI report)*, a BI user report must first exist in Oracle CX Sales. An example of a report path is `/Custom/usyncrep.xdo`

- **Eloqua Object:** Select *User*.
- Click **Fetch** to see available fields for mapping.

 **Note:** The **Fetch** button is only available for Oracle CX Sales (formerly Oracle Sales Cloud) version R12 or higher. Older versions of Oracle CX Sales automatically load fields to be mapped.

- **Update Eloqua Fields:** Map your unique identifier to specify the object in Oracle Eloqua to update or create. For each set of fields, select the Oracle Eloqua field that will be mapped to Oracle CX Sales contacts. Add more mappings by clicking **Add Import Field Mapping**. Click the menu  to delete the mapping or switch the field type to static text, static date, or dynamic date.



- **Import Rules:** Specify folder and deduplication rules.

**Import Rules**

Folder: Users / SC\_Users\_Import

Deduplication Type: Ignore all duplicates and do not import

Specify how to handle duplicate import records

- **Folder:** Select an Oracle Eloqua folder where the user should be saved. All existing user folders in your instance are available in this drop-down menu. New folders cannot be created from this menu.
- **Deduplication type:** Specify how to handle duplicate import records. For example, you can choose the most recently updated record or import all records. Deduplication runs per import execution. If multiple records with the same match field value are imported in a single execution, those records will be deduplicated based on the setting specified here.
- **Post Processing:** Click **Add Action** to specify an action to be performed on a user after import.

**Post Processing**

Specify actions to be performed on a contact after import

Add to Security Group

Add to Security Group

Remove From Security Group

Select...


- **Add to Security Group / Remove from Security Group:** Specify whether after the import, users should be added to or removed from a security group.

#### 4. Click **Save**.

Click the drop-down list to modify the import. You can edit, enable (to schedule your import) and disable, or view report data for an import. If an import is disabled, you can delete it.


Once your import is scheduled, it automatically runs every 15 minutes. The import can be run on demand using the **Run Now** option. When **Run Now** is used with a date and time filter, the current date and time is pre-populated. The date and time can be changed to a time in the past as desired. If no date is selected, the import retrieves all matching records from Oracle CX Sales and imports them to Oracle Eloqua. Selecting a specific date

allows you to recover missed records and perform partial imports if field mapping changes.

 **Note:** The *Last Import* date indicates the last successful import.

Clicking **Disable** temporarily stops the imports. If the import is currently running and has not yet started syncing to Oracle Eloqua (if it is in either the *Retrieve from CX Sales* or *Deduplicating* steps), the import is canceled and disabled. If the import has begun importing to Oracle Eloqua, the import will finish, then be disabled. Click **Enable** to enable your import.

## Configuring notifications

 **Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

An import or action may fail due to various reasons such as connection or data related issues. Use the *Notifications* page to setup email notifications to alert you of import and action errors. The notification includes your client and import name, start and end times, and identifies the step at which the error occurred.

## ✖ An error has occurred with your import from CRM

ERROR: Retrieving from Sales Cloud

### Import details:

|             |                          |
|-------------|--------------------------|
| Client:     | ExampleClient            |
| Import:     | TestImport               |
| Start Time: | 2016-11-24T22:01:00Z     |
| End Time:   | 2016-11-24T22:01:02.224Z |

If the problem persists, please log into [My Oracle Support](#) and submit a case or call 1-800-223-1711 for technical support.

We apologize for the inconvenience.

### To configure a notification:

1. In the Oracle CX Sales Integration app, click **Notifications**.
2. Click **Add Notification**.
3. Select the notification settings and recipients:

#### Add Notification

|              |   |
|--------------|---|
| Name         | <input type="text" value="Action Notifications for Test Connection"/> |
| Connection   | <input type="text" value="Test Connection Group"/>                    |
| Event Source | <input type="text" value="Actions"/>                                  |
| Action       | <input type="text" value="All Actions"/>                              |
| Event Type   | <input type="text" value="Failure"/>                                  |
| Email(s)     | <input type="text" value="marzena@oracle.com,katie@oracle.com"/>      |

Separate multiple addresses with a comma ","

- **Name:** Provide a descriptive name that will appear in the list of notifications.
- **Connection:** Select a specific connection or all connections.

- *Event Source*: Notifications are currently available for imports and actions.
- *Import or Action*: Select the import or action, or all events, for which you want to receive notifications.
- *Event Type*: Notifications are available for failed imports and actions.
- *Email(s)*: Enter recipient email addresses. You can enter multiple email addresses separated by commas.

4. Click **Save**.

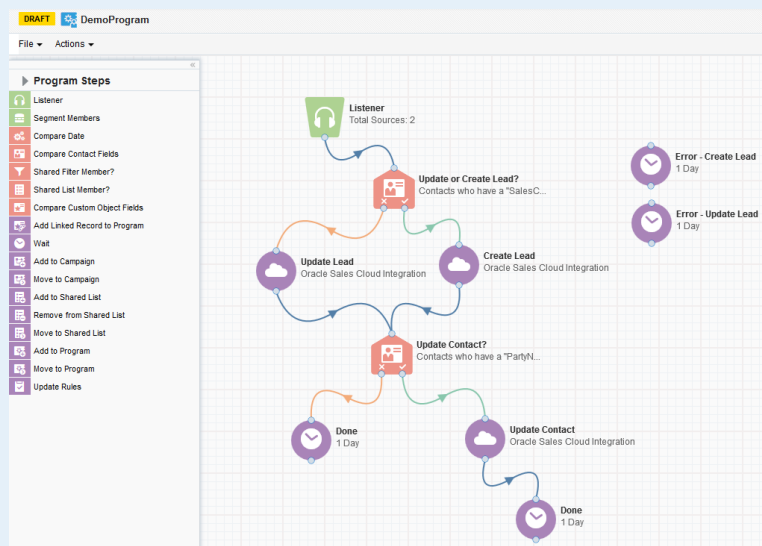
You can return to the notification area to create new notifications, as well as edit, delete, and disable existing notifications.

# Using the Oracle CX Sales app in a program or campaign

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

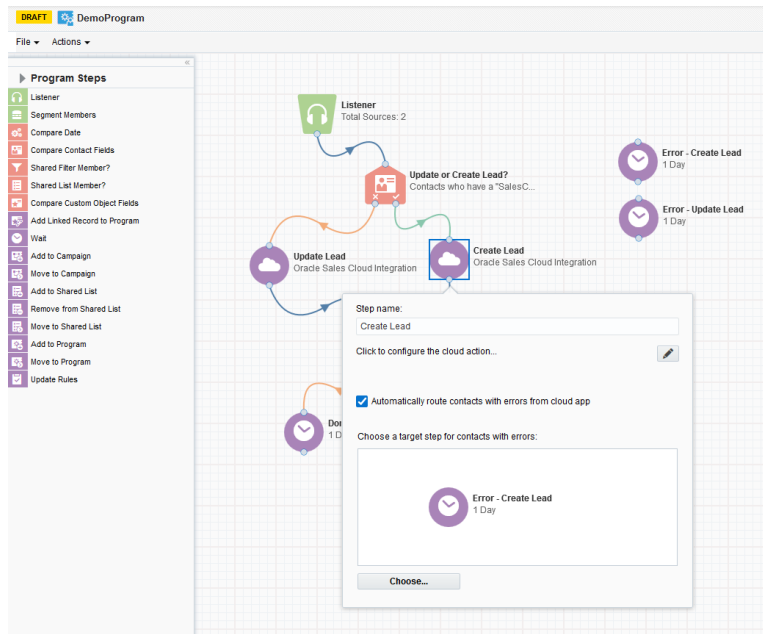
The Oracle CX Sales Integration app enables you to update or create contact records within a program or campaign.

**Example:** A listener step could be configured to listen for contacts who complete a form. Once the contact is added to the [program canvas](#), you may want to decide to create a lead for the contact, depending on the next decision, you can configure this app to either create a contact or update an existing contact within the CRM.



To use the Oracle CX Sales Integration app in a campaign or program:

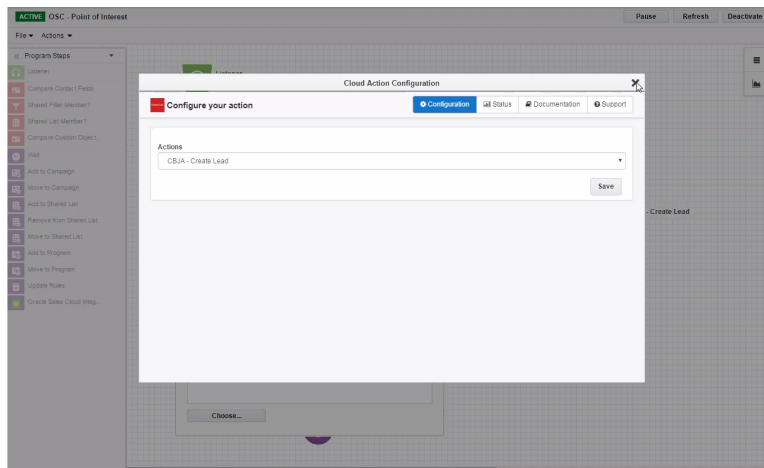
1. Open an existing campaign or program, or create a new one.
2. Click on the *Actions* section to expand it, and view all available action elements, including installed apps.
3. Drag the **Oracle CX Sales Integration** element onto the canvas, then double-click it to open the configure screen.



The window that opens enables you to do the following:

- Change the name of the step to something more meaningful.
- Configure the element.
- Choose to route contacts that resulted in an error to another step (see: [configuring campaign elements](#) for more information).

4. Click the **Edit** icon to configure the cloud action.



- a. Click the **Actions** list and select an existing action. If you do not see any existing actions, see [configuring actions](#) for more details.
- b. Click **Save**.
- c. (Optional) Click **Status** to see the status of contact processing. The amount of contacts that have been successfully processed, or were processed with warnings or errors will be displayed.

# Viewing report data in the Oracle CX Sales Integration app

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

The Oracle CX Sales Integration app provides reporting on the records being synced between Oracle Eloqua and Oracle CX Sales to provide insight into how your syncs are performing. Data is retained for a period of 3 months. The Oracle CX Sales Integration app provides reporting in two different formats:

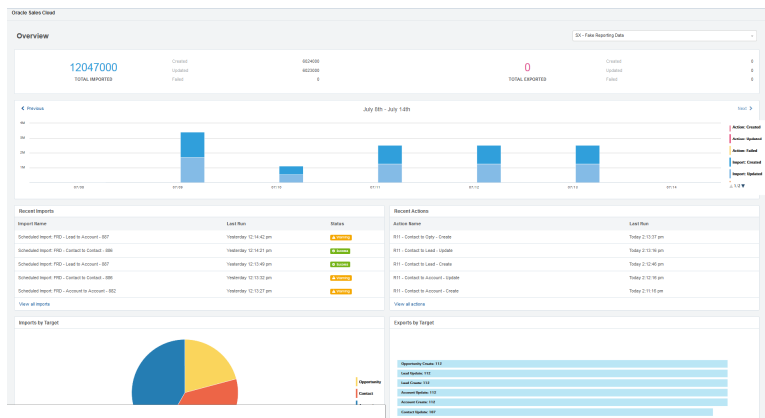
- A [Global Report Dashboard](#) which displays import and action activity for the past seven days.
- Individual [reporting for imports and actions](#) that displays activity for a specific import or action for the current day (previous dates are selectable). A table of every import or action execution, including details of created, updated, and failed records and time of execution.

## Viewing the global report dashboard in a cloud menu

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.



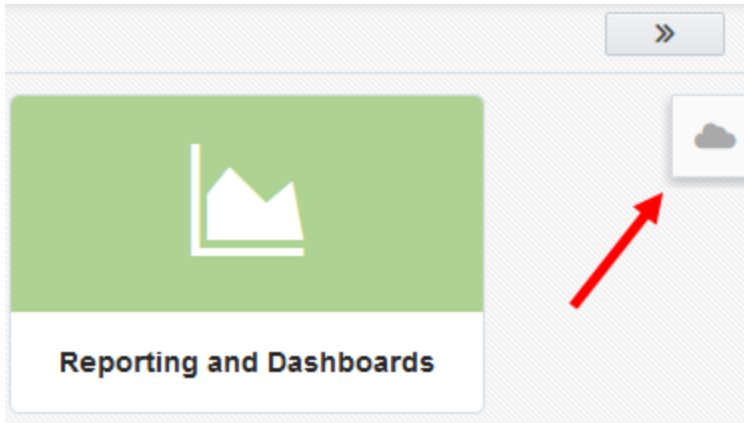
The Oracle CX Sales Integration Reporting Dashboard provides an overview of action and import activity over a span of the last seven days. In this report, the volume of data flowing between Oracle Eloqua and Oracle CX Sales is shown per day, including created, updated, and failed records. The report initially displays the most recent seven days of data, with an option to select a specific seven day period in the past.



The global report dashboard is a cloud menu. Data is retained for a period of 3 months.

## To view the global report dashboard:

1. In Oracle Eloqua, you can access this report from a cloud menu in a campaign, program, or My Eloqua.
2. Click **Cloud Content** to open the Cloud Content menu, which lists all available content services.

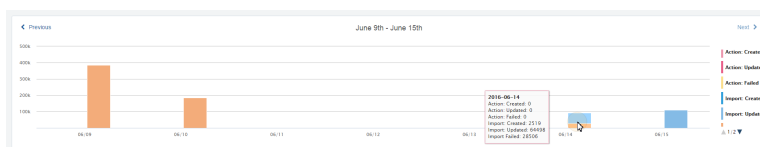


3. Locate the **Oracle CX Sales Integration Reporting Dashboard**, either by scrolling through the list or searching with the search box.
4. Click the menu to open the dashboard.

Select a CRM connection for which you want to view a report. An overview of the total amount of imports and exports over the current seven day period that were created, updated, or failed are displayed.



You can view the amount of records created, updated, or failed over a seven day period with an option to view the details for each specific day by mousing over the day. Here, you can view reports for a specific seven day period in the past.



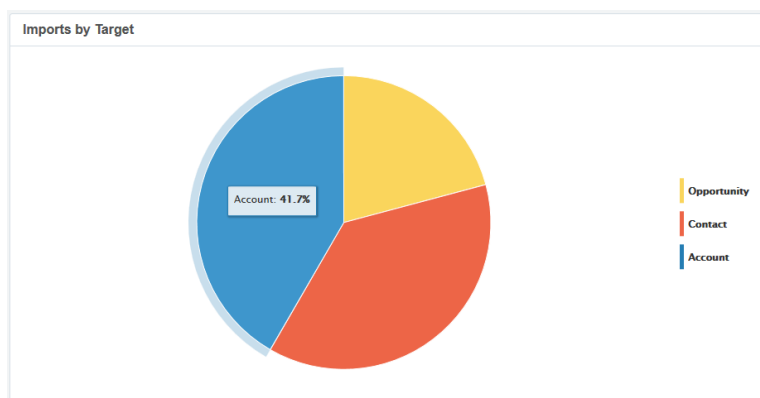
## Viewing recent imports and actions

The report also displays recently successful imports and actions and the date and time they were last successful.

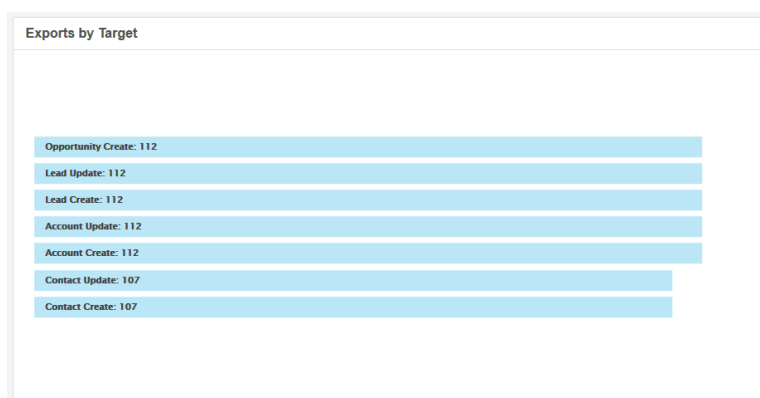
| Recent Imports   |                  |         |
|--|------------------|---------|
| Import Name  | Last Run         | Status  |
| Scheduled Import: Mocked Sales Cloud Connection - SANITY IMPORT - v2 | Today at 2:52 PM | Success |
| Scheduled Import: Mocked Sales Cloud Connection - SANITY IMPORT - v2 | Today at 2:52 PM | Success |
| Scheduled Import: Mocked Sales Cloud Connection - SANITY IMPORT - v2 | Today at 2:52 PM | Success |
| Scheduled Import: Mocked Sales Cloud Connection - SANITY IMPORT - v2 | Today at 2:52 PM | Success |
| Scheduled Import: Mocked Sales Cloud Connection - SANITY IMPORT - v2 | Today at 2:52 PM | Success |
| <a href="#">View all imports</a>                                     |                  |         |

## Viewing imports and exports by target

Import data is categorized and displayed in a pie chart to help visually identify how many records are being imported into Oracle Eloqua from Oracle CX Sales by target.



The *Exports by Target* bar chart displays all records within the period so you can view how many records were created or updated from Oracle Eloqua to Oracle CX Sales, categorized by target.



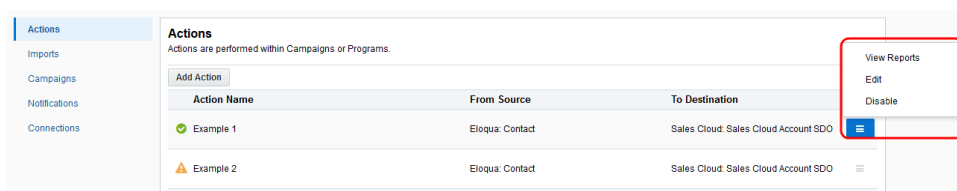
# Viewing report data for actions and imports

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

Individual reporting for Oracle CX Sales Integration app actions and imports displays activity for a specific import or action for the current day (previous dates are selectable). A table of every import or action execution, including details of created, updated, and failed records and time of execution. Data is retained for a period of 3 months.

## To view report data for actions or imports:

1. In the Oracle CX Sales Integration app, click **Actions** or **Imports**.
  - a. To view reports for [campaign actions](#), click the **Campaign Actions** tab.
2. Locate the action or import for which you want to view report data, click the list and select **View Reports**.

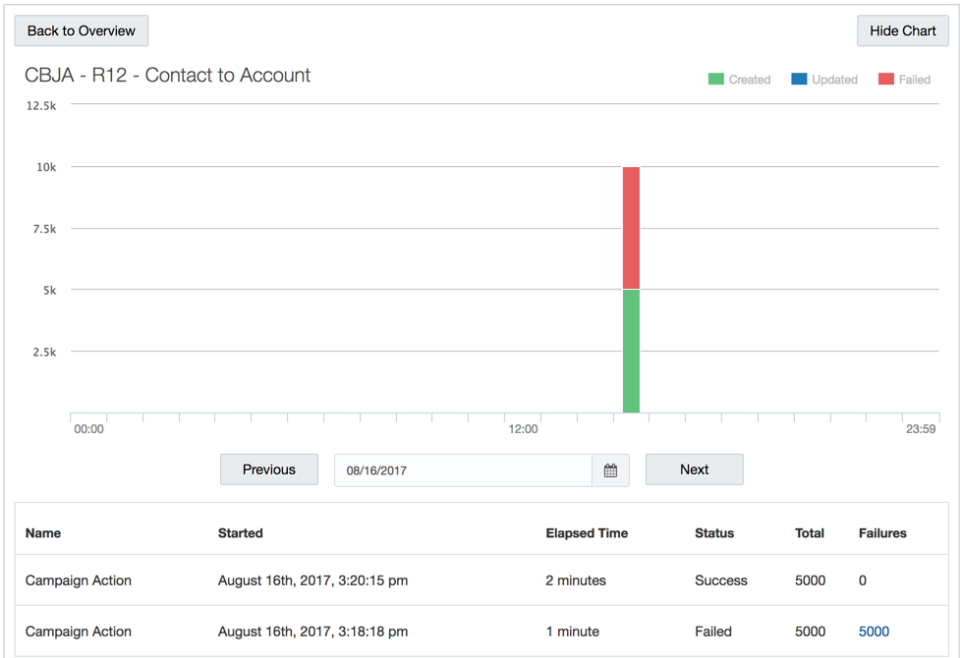


## Action report data

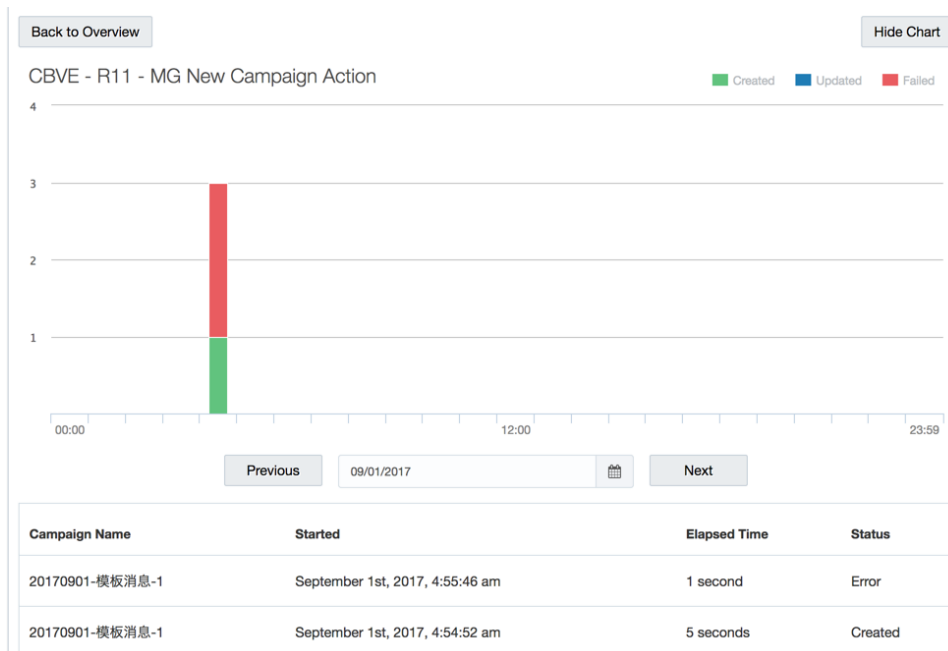
Action reports are available for step actions and campaign actions. These reports provide useful information about the records that are flowing from Oracle Eloqua into Oracle CX

Sales.

For step actions, view how many records were created or updated for each sync, how many were processed per hour, how long the sync took to complete, and whether or not the sync was successful.



For campaign actions, view whether the campaigns exported from Oracle Eloqua to CRM were successful and a daily history of your campaign exports.

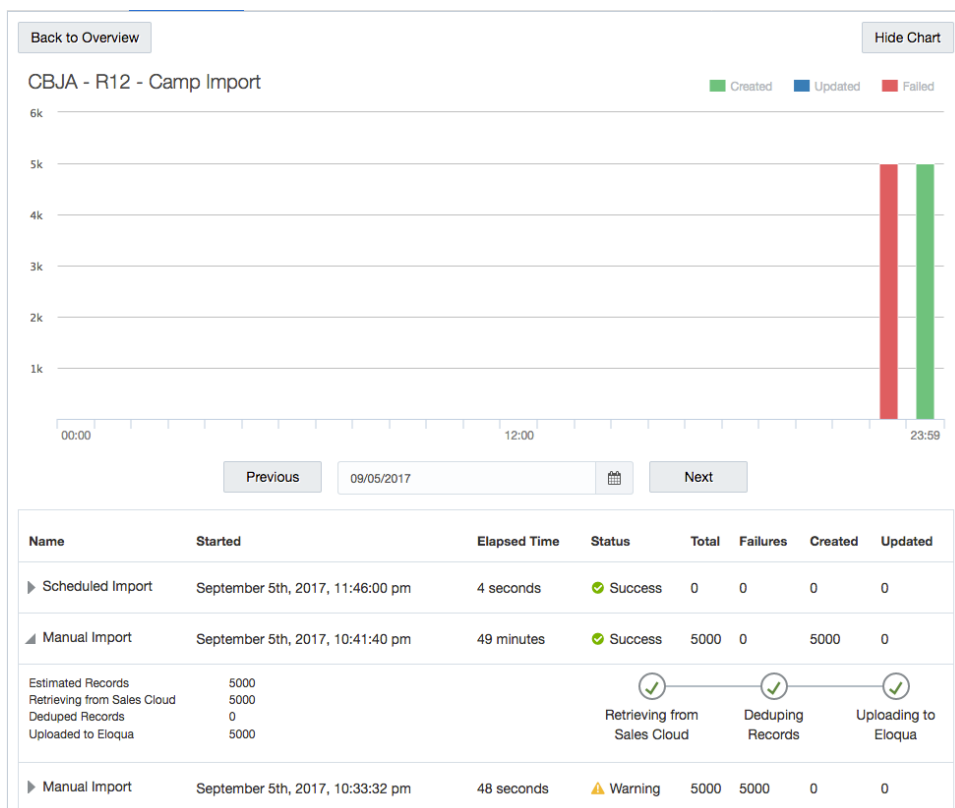
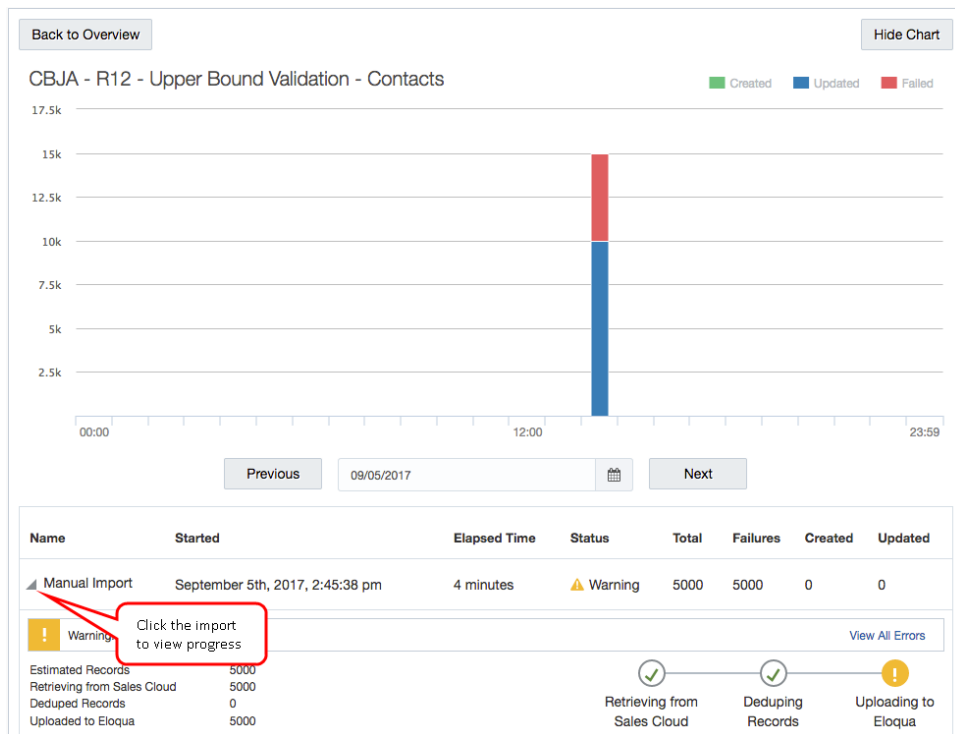


Click a table row to view more details of the data sent over, sent date, and error messages.

## Import report data

Import reports provide insight into how your imports are performing by displaying a chart of how many records are being created, updated, or have failed to import. You can also view how many objects were imported into Oracle Eloqua campaigns from CRM, which ones were successful, and how many failed.

Each report displays an import schedule of when syncs started, how long the sync took to complete, whether or not that sync was successful, and how many records were created, updated, or failed during that sync. When an import is running, click the import to view the progress.



The status of the import is displayed including:

- Import data
- When the import started
- How long the import took to complete
- The number of records with their status

There are three stages to a successful import; records are retrieved from Oracle CX Sales, deduplicated, and uploaded to Oracle Eloqua. The message *Retrieving from CX Sales* indicates your import is successfully running.

You can manually stop an import if it is running and has not yet reached the *Uploading to Eloqua* step. Once the *Uploading to Eloqua* step is started, the import cannot be stopped and will continue to completion. Stopped imports cannot be resumed. You must run the import again. Learn more about [creating and scheduling imports](#).




# Example programs

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

You can use the Oracle CX Sales Integration app for various marketing qualification processes and lead generation. You can use Oracle Eloqua's program canvas to design programs that will update and create leads and contacts in Oracle CX Sales based on the following recommended programs:

- **Create Unique:** In this program, a new lead record is created in Oracle CX Sales only if there is no contact or lead record with the same email address in Oracle CX Sales. If there is an existing contact record, the contact record is updated with contact information as defined in the *Update Contact* action.
- **Hybrid:** In this program, a new lead is created in Oracle CX Sales only if there is no unconverted lead record with the same email address in Oracle CX Sales. Existing unconverted lead records in Oracle CX Sales are updated with contact information as defined in the *Update Contact* action.
- **Point of Interest:** A new lead record is created if there are no leads with the same email address in Oracle Sales, but an Oracle CX Sales contact already exists. If there is an existing contact record, the contact record is updated with contact information from Oracle Eloqua as defined in the *Update Contact* action.
- **Automatic Opportunity Qualification Program:** The Automatic Opportunity Qualification Program is a discretionary part of Oracle Guided Campaigns, and it is designed to automatically create new Sales Leads and Opportunities based on responses in account-based marketing

campaigns. Currently, related data flows are supported in the context of Eloqua CX Sales CRM integration. For more information, please refer to Oracle Guided Campaigns.

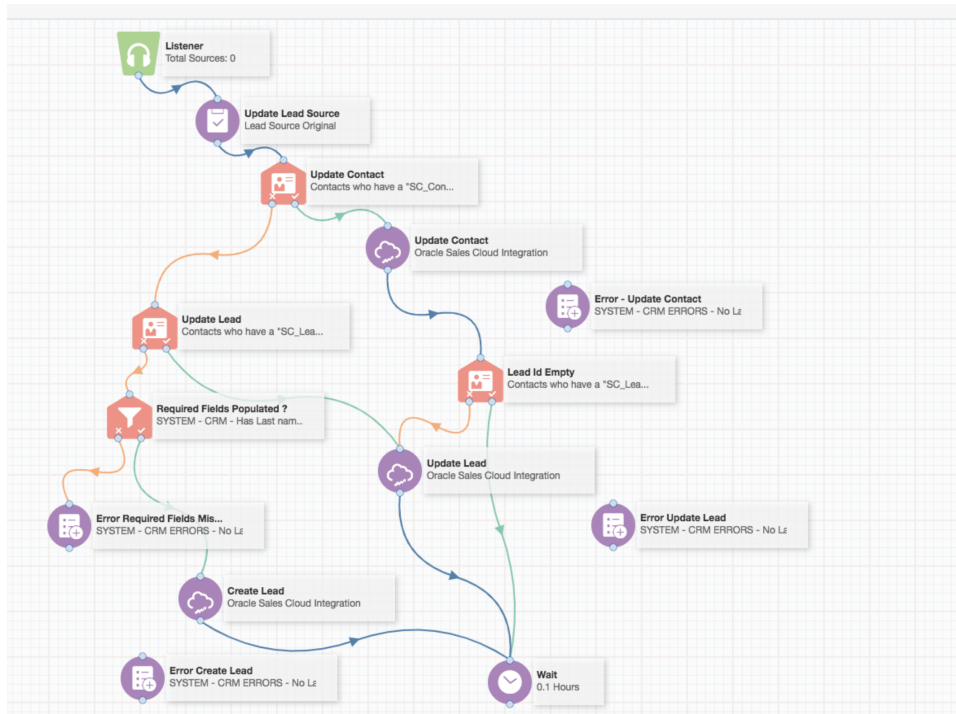
 **Tip:** You can [select default assets to automatically create programs](#).

## Creating a create unique program

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

You can use Oracle Eloqua's [program canvas](#) to manually configure a create unique program or you can [select default assets to automatically create programs](#).



Default create unique program:

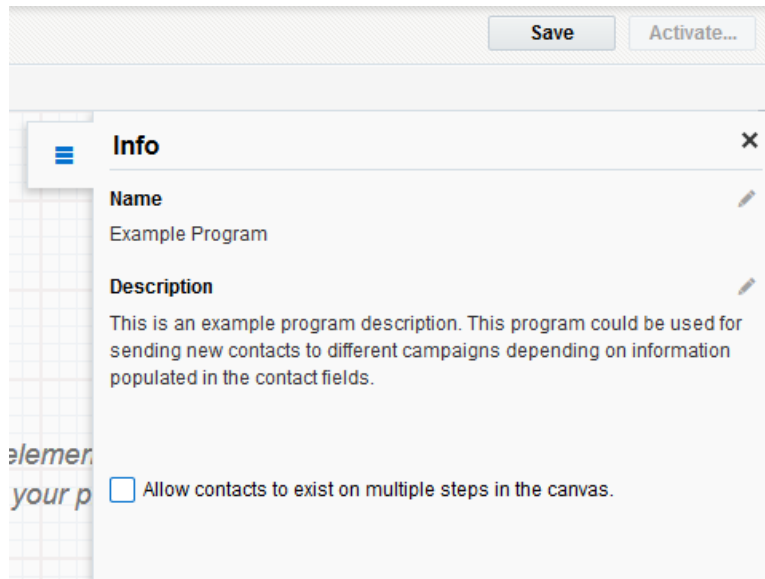



In this program, a new lead record is created in Oracle CX Sales only if there is no contact or lead record with the same email address in Oracle CX Sales. If there is an existing contact record, the contact record is updated with contact information as defined in the *Update Contact* action.

**Note:** Determining whether an Oracle CX Sales (formerly Oracle Sales Cloud) contact or lead exists depends on configuring an import for both Oracle CX Sales leads and contacts with the appropriate record IDs. For details, see [recommended contact and account fields](#).

**To create an Oracle CX Sales Create Unique program:**

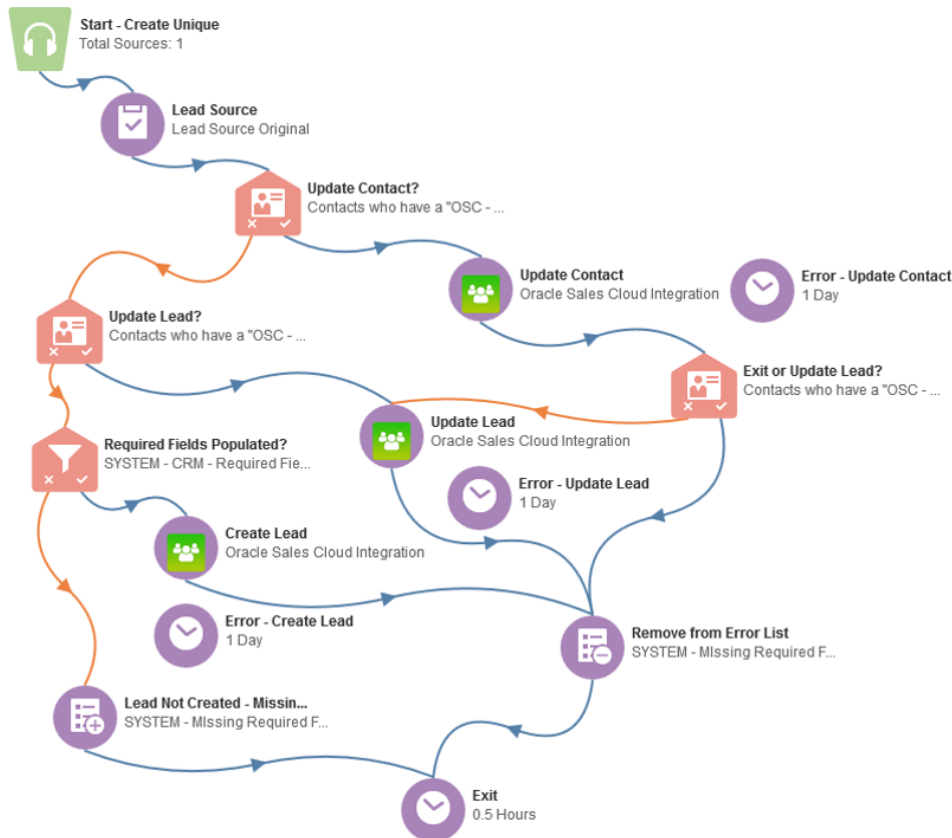
1. Navigate to **Orchestration** , then click **Programs**.
2. Click **Create a Contact Program**
3. Select a blank template.
4. Configure basic information about the program:
  - a. Click **Info**  on the left side of the canvas, and add a **Name** and **Description** of the program.



- b. To allow a contact to enter the program more than once, select **Allow contact to exist on multiple steps in the canvas**. If you do not select this option, once a contact enters a program, the contact cannot enter the program again from another entry point.
    - c. Click **X** to close the *Info* window.
5. Add the **Listener Step** to the canvas:
  - Name the step *Start - Create Unique*.
  - To subscribe to data using a listener step, double-click  **Listener Step**.
  - Add any data sources that are part of your lead flow. For example, form submissions, lead nurturing programs or campaigns that generate leads.
6. Add an **Update Rule** action step. Name the step *Lead Source*. Configure the *Rule Set* option to **Lead Source Original**.

7. Add a **Compare Contact Fields** decision step. Name the step *Update Contact?* This step evaluates and routes contacts based on a specified value in a selected contact field. Set the step to route contacts based on whether they have an *Oracle CX Sales Contact Registry ID*.
  - a. For the *yes* path, add a **Oracle CX Sales Integration** action to update information for existing Oracle CX Sales contacts. Name this step *Update Contact*.
  - b. For the *no* path, add a **Compare Contact Fields** decision to route contacts based on whether they have an *Oracle CX Sales Lead ID*. Name this step *Update Lead?*
8. For the *Update Lead?* step, add the following paths:
  - a. For the *yes* path, add a **Shared Filter Member?** action. Name this step *Required Fields Populated?*. This step evaluates whether shared filter members have required contact fields (such as name, company, and so on). Add an **Oracle CX Sales Integration** action named *Create Leads* and connect it to the *yes* path to create leads for filter members with the required contact fields.
  - b. For the *no* path, add an **Oracle CX Sales Integration** action to update information for existing Oracle CX Sales leads. Name this step *Update Lead*.
9. Add any necessary wait steps, error flows, shared list actions, and Oracle CX Sales app actions as depicted in the workflow.
10. Click **Save**.
11. Activate your program.
12. Monitor the performance of the program using operational reports. If you need to change your program, you must disable the program.

The following example canvas illustrates a manually-configured create unique program:



## Creating a hybrid program

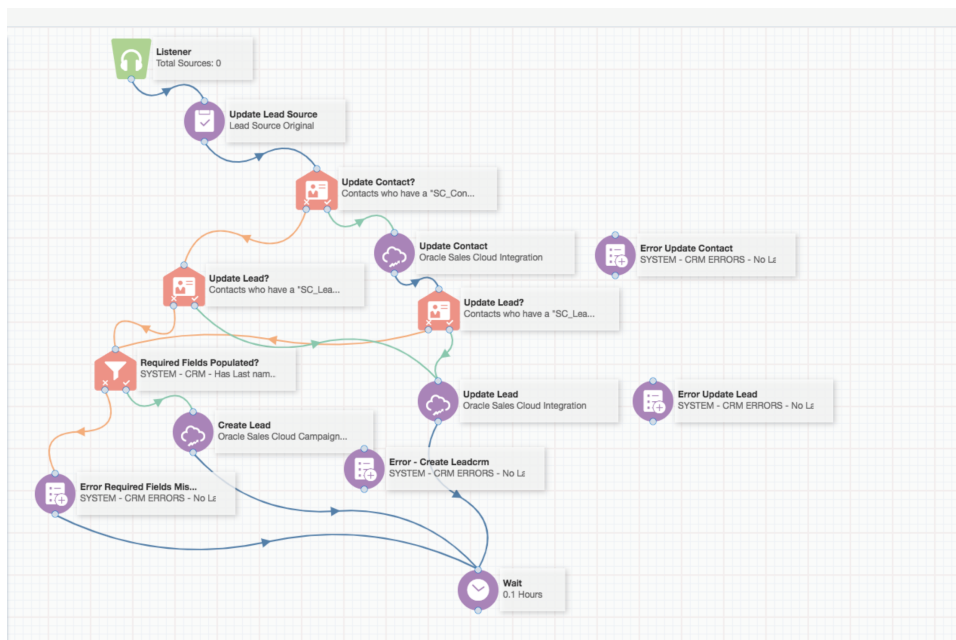
**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

A hybrid program creates new leads and updates existing contacts and leads in Oracle Sales. If leads with the same email address do not yet exist, the hybrid program determines whether the contacts are valid CRM contacts by checking certain fields (such as last name or company name) before creating a lead in the CRM system.

**Note:** Determining whether an Oracle CX Sales (formerly Oracle Sales Cloud) contact or lead exists depends on configuring an import for both Oracle CX Sales leads and contacts with the appropriate record IDs. For details, see [recommended contact and account fields](#).

You can use Oracle Eloqua's [program canvas](#) to manually configure a hybrid program or you can [select default assets](#) to automatically create hybrid programs.

Default hybrid program:

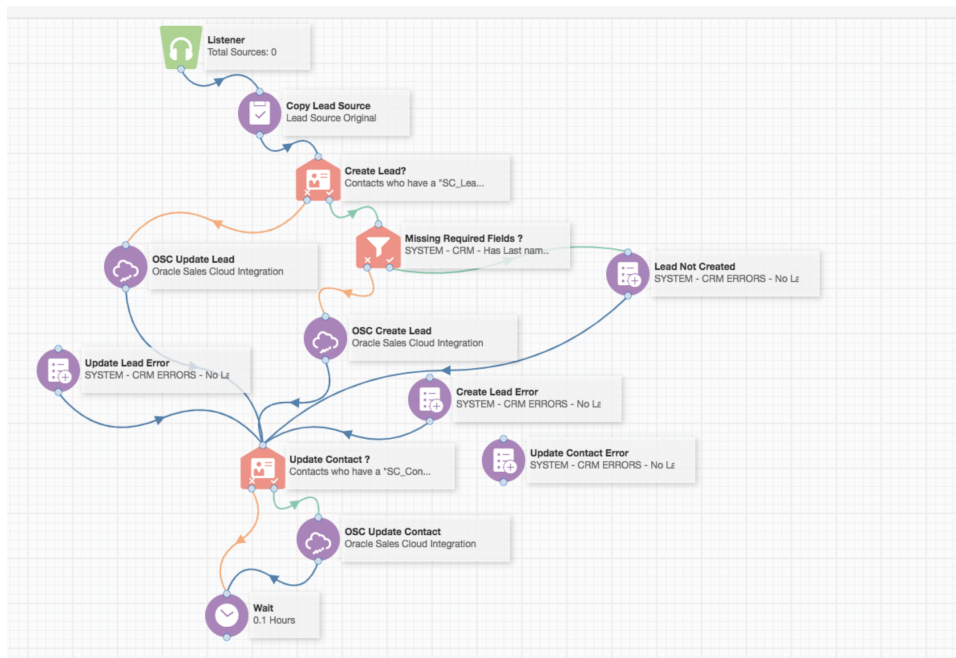


## Creating a point of interest program

**Important:** *Oracle Sales* was formerly known as *Oracle Sales Cloud*. All instances of Oracle Sales Cloud have been changed to Oracle Sales across the app user interface.

You can use Oracle Eloqua's [program canvas](#) to manually configure a point of interest program or you can [select default assets to automatically create programs](#).

Default point of interest program:





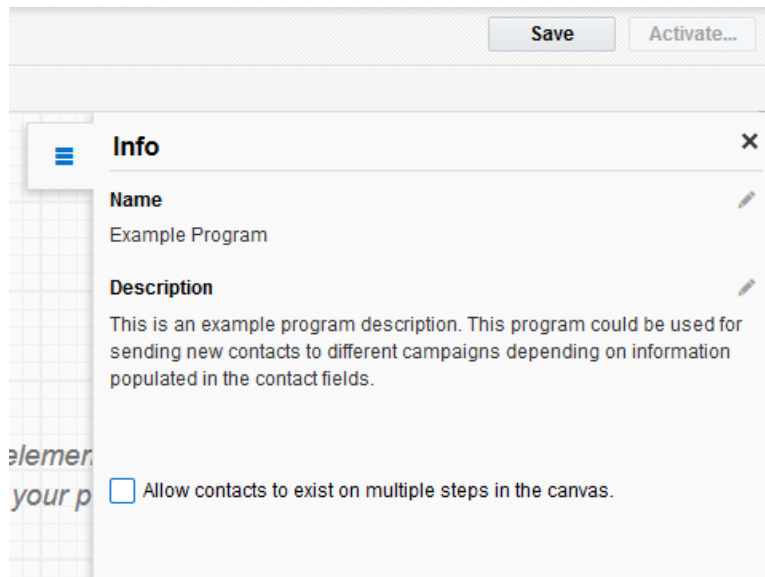
A new lead record is created if there are no leads with the same email address in Oracle Sales, but an Oracle CX Sales contact already exists. If there is an existing contact record, the contact record is updated with contact information from Oracle Eloqua as defined in the *Update Contact* action.



**Note:** Determining whether an Oracle CX Sales (formerly Oracle Sales Cloud) contact or lead exists depends on configuring an import for both Oracle CX Sales leads and contacts with the appropriate record IDs. For details, see [recommended contact and account fields](#).


### To create an Oracle CX Sales point of interest program:

1. Navigate to **Orchestration** , then click **Programs**.
2. Click **Create a Contact Program**
3. Select a blank template.
4. Configure basic information about the program:
  - a. Click **Info**  on the side of the canvas, and add a **Name** and **Description** of the program.




Save Activate...

**Info** ×

**Name** 

Example Program


**Description** 

This is an example program description. This program could be used for sending new contacts to different campaigns depending on information populated in the contact fields.

☐ Allow contacts to exist on multiple steps in the canvas.

- b. To allow a contact to enter the program more than once, select the **Allow contact to exist on multiple steps in the canvas** check box. If you do not select this option, once a contact enters a program, the contact cannot enter the program again from another entry point.
- c. Click **X** to close the *Info* window.

5. Add the **Listener Step** to the canvas:

- Name the step *Start - Point of Interest*.
- To subscribe to data using a listener step, double-click  **Listener Step**.
- Add any data sources that are part of your lead flow. For example, form submissions, lead nurturing programs or campaigns that generate leads.

6. Add an **Update Rule** action step. Name the step *Lead Source*. Configure the *Rule Set* option to **Lead Source Original**.

7. Add a **Compare Contact Fields** decision step. Name the step *Create a Lead?* This step evaluates and routes contacts based on a specified value in a selected contact field. Set the step to route contacts based on whether they have an *Oracle CX Sales Lead ID*.

- a. For the *yes* path, add a **Shared Filter Member?** action. Name this step *Are Required Fields Populated?*. This step evaluates whether shared filter members have required contact fields (such as name, company, and so on). Add an **Oracle CX Sales Integration** action named *OSC - Create Lead* and connect it to the *yes* path to create leads for filter members with the required contact fields. For the *no* path, move members to an **Add to Shared List** action named *Lead Not Created - Missing Fields*.
- b. For the *no* path, add an **Oracle CX Sales Integration** action to update information for existing Oracle CX Sales leads. Name this step *OSC - Update Lead*.

8. Add a **Compare Contact Fields** decision step. Name this step *Update Contact?*

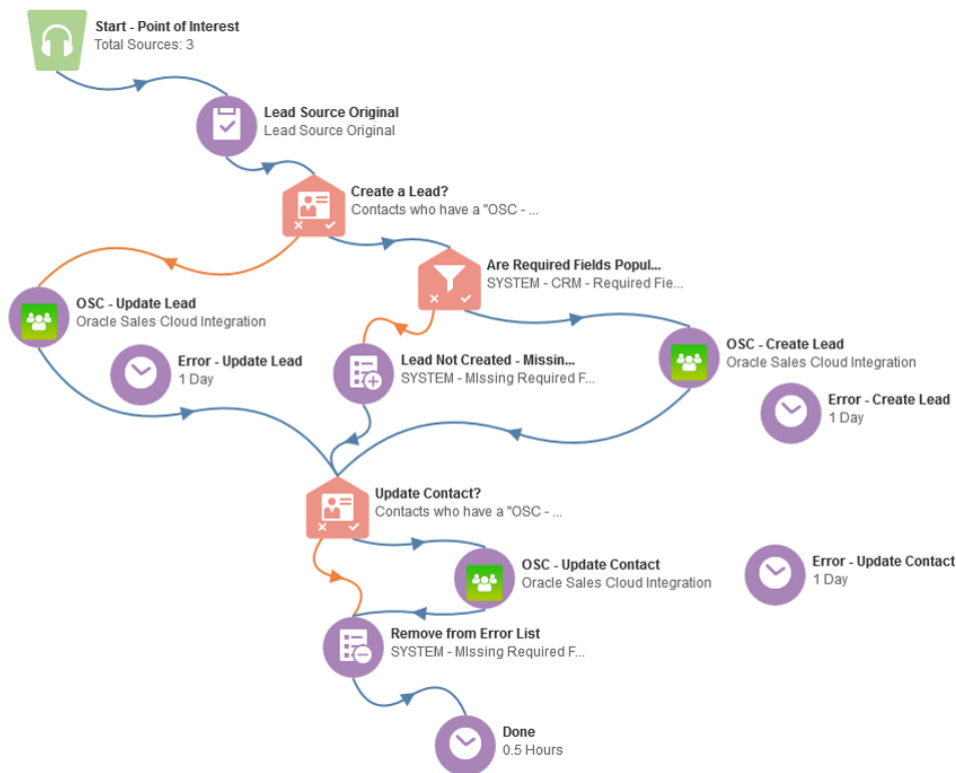
- a. For the *yes* path, add an **Oracle CX Sales Integration** action to update information for existing Oracle CX Sales contacts. Name this step *OSC - Update Contact*. Connect this step to the *Remove from Error List* step below.
- b. For the *no* path, add a **Remove from Shared List** action. Name this step *Remove from Error List*.

9. Add any necessary wait and error steps.

10. Click **Save**.

11. Activate your program.
12. Monitor the performance of the program using operational reports. If you need to change your program, you must disable the program.

The following example canvas illustrates a manually-configured point of interest program:



## Create an Opportunity Qualification Program

**Tip:** These instructions are designed for your Eloqua administrator to setup and configure your automated CRM Integration program, so that it will thereafter be available for selection within Oracle Eloqua or Oracle Guided Campaigns for the purposes of [Automated Opportunity Qualification](#).

The Automatic Opportunity Qualification Program is a discretionary part of Oracle Guided Campaigns, and it is designed to automatically create new Sales Leads and Opportunities based on responses in account-based marketing campaigns. Currently, related data flows are supported in the context of Eloqua CX Sales CRM integration. For more information, please refer to Oracle Guided Campaigns.

Prior to starting your CRM Integration set-up you need the following items enabled within your Oracle Eloqua instance.

- Decision Steps
- Upsert Actions
- Marketing Activity Sync
- Eloqua Campaign Lead Table

To enable these you should raise an Service Request with Oracle Support requesting they enable Guided Campaigns integration features.

## CRM Integration program overview

The CRM Integration program needs to be created manually in your Oracle Eloqua instance. It is necessary to include a listener as the input (so that you can push contacts to it from your other programs, or your Oracle Guided Campaigns campaigns). It will also be necessary to include Oracle Sales decision steps and Oracle Sales action steps to validate your contacts for processing.

### Integration setup

The integration flow will primarily consist of two elements, the decision steps which will reference your Oracle Sales instance), and the actions which will enact the data synchronization with your Oracle Sales instance.

**Note:** All decision and action steps need to be set-up and mapped by your Eloqua Administrator prior to your program setup.

## Program actions

Users of Oracle Guided Campaigns can utilize an Account Engagement Score Decision step to evaluate threshold based uniquely on Campaign and Account. there is also a Unity Decision step as part of the Unity Integration App that allows you to score based on your unity data, uniquely by Product and Account.

The program flow will consist of 3 types of actions that need to be configured.

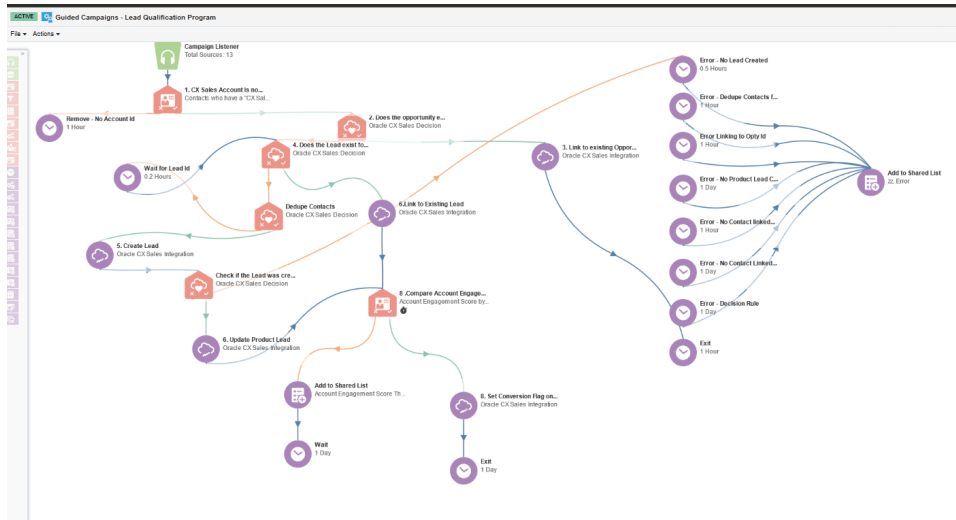
1. [Eloqua native program decision steps](#) - these reference the Eloqua contact record.
2. [Oracle CX Sales Integration app decision steps](#) - these reference additional Eloqua database tables or the Oracle Sales data housed within the Eloqua database.
3. [Oracle Sales Action steps](#) - these are either create, update or upsert calls made to your Oracle Sales instance or additional back-end Eloqua integration database tables.

## Imports

There is one type of Import that needs to be configured:

1. [Opportunity Import](#) - The purpose of this import is to sync the Opportunity ID created in your Oracle Sales instance to pull it back to Eloqua for association to Leads created within the campaigns.

Below is a diagram showing the template flow for the integration program:



**Note:** The step numbers below all relate to ones on the above diagram as an illustration.

## CRM Integration program setup

### Listener)

A listener added to your program will make it feasible to drive contacts into the program from other sources, this is necessary if you wish to add contacts in from your campaigns created within Oracle Guided Campaigns.

The program listener is required as your first step to facilitate the receipt of contacts from your other sources, such as campaigns created in Oracle Guided Campaigns.

Guided Campaigns feeder  
Total Sources: 1

Step name:  
Guided Campaigns feeder

Total Sources: 1

+ View Sources

Native Eloqua Decision steps

### Step 1: CX Sales Account is not blank

This Eloqua decision step validates whether CRM Account ID is blank. A CRM Account ID needs to be present in your chosen Eloqua field for the Opportunity flow to work as needed.

1. CX Sales Account is not blank?  
Contacts who have a "CX Sales Account Id" field havir

Step name:  
1. CX Sales Account is not blank?

Evaluate Contacts whose...  
CX Sales Account Id

☒ not ☐ is blank

Evaluation period: 0 Day(s)

### Step 8: Compare Account Engagement score to 'n'

This Eloqua decision step references the Account engagement score by campaign to compare against your chosen value. The Account Engagement Score by campaign Decision Step is made available to all Oracle Guided Campaigns customers.

8 .Compare Account Engagement Score < 20  
Account Engagement Score by Campaign is less than 20

Step name:  
8 .Compare Account Engagement Score < 20

Contacts whose Account Engagement Score by Campaign is  
not less than  
20

Evaluation period: 1 Day(s)

lag on...  
gration

(You can define the score thresholds that you wish to use, as well as the evaluation period)

• If you have Oracle Unity you can also use the Unity decision step to get scores from your Unity instance, which is unique by Product and Account

### Oracle Sales Integration app decision steps

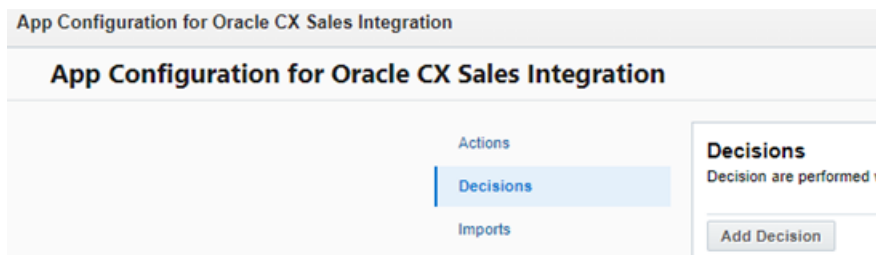
Oracle CX Sales decision steps use decision rules setup in the Oracle CX Sales Integration app under Decisions. After the rules are setup in the app, you reference them from the integration program using the Oracle CXSales decision step.

The decision steps will reference the records in your Oracle Sales instance or additional data stored in Eloqua to validate that the record has values where necessary, and that those values are valid as far as the requirements for your integration.



**Note:** Oracle CXSales decision steps are currently only available to Oracle Guided Campaigns customers. For more details on this, see the [Oracle CX Sales Integration app Help Center](#).

Decision rules are set within the CX Sales Integration app under **Decisions**.



## Step 2: Does the Opportunity Exist?

The decision rule is a filter rule that will query the CampaignOpportunity ID along with the CRM Campaign ID referenced from the campaigns. The step needs to validate if that data is blank or not. (if NOT is checked, then value presence will result to true and flow down the yes path.)

The image shows the 'Edit Decision' form. At the top, there are 'Cancel' and 'Save' buttons. Below them is a description: 'Control the path Eloqua contacts take as they flow through this Canvas Step. Filtered records will be pushed to the "Yes" path, while the remaining records will move to the "No" path in a canvas.' The form contains several fields: 'Decision Name' with the value '2 - Does the opportunity exist for Campaign?', 'Eloqua Object' set to 'Contact', 'Connection Name' set to 'CBVE', and 'Filter Rules Purpose' with 'Filter' selected (radio button). Below this is a section titled 'Filter Rules' with the instruction 'Use Contact Fields to define a rule'. A rule is defined with the field 'CampaignLead Cnn Opportunity Id', the operator 'not' (checked), and the condition 'Is Blank'. There is an 'Add Filter Rule' button at the bottom of the rule section.

## Step 4: LeadID is not blank.

The decision step is a filter rule that will query the CampaignLead ID along with the CRM Lead ID and the CRM Campaign ID referenced from the campaigns.

The step needs to validate if the field is blank or not. (if NOT is checked, then value presence will result to true and flow down the yes path.) The purpose of this is to ensure that only one Lead is created for an Account (per campaign).

**Edit Decision** Cancel Save

Control the path Eloqua contacts take as they flow through this Canvas Step. Filtered records will be pushed to the "Yes" path, while the remaining records will move to the "No" path in a canvas.

Decision Name: 4 - Lead ID is not blank

Eloqua Object: Contact

Connection Name: CBVE

Filter Rules Purpose: ☒ Filter ☐ Dedupe

**Filter Rules**

Use Contact Fields to define a rule

CampaignLead Cn Campaign Lead Id ☒ not is Blank

Add Filter Rule

## Dedupe contacts

This decision step will deduplicate to ensure that only a single lead is created per account (per campaign). This dedupe rule deduplicates by Company name + Campaign ID, to ensure that per processed batch from a single campaign, only one contact for each account is routed down the yes path.

**Edit Decision** Cancel Save

Control the path Eloqua contacts take as they flow through this Canvas Step. Filtered records will be pushed to the "Yes" path, while the remaining records will move to the "No" path in a canvas.

Decision Name: Dedupe Contacts by Account and Campaign - 23.02

Eloqua Object: Contact

Connection Name: CBVE

Filter Rules Purpose: ☐ Filter ☒ Dedupe

**Dedupe Rules**

Use Contact Fields to define a rule

Advanced Mode ? Insert Object

SourceCampaign.Id

And

Add Dedupe Rule

## Actions

**Note:** For more information on creating actions, see the [Oracle CX Sales Integration app Help Center](#).

### Step 5: Create Lead

This rule requires a create Lead call. For the core mapping that you wish to set up, Campaign ID **must** be included on this call.

The screenshot displays a mapping configuration interface for the 'Create Lead' action. It features a grid of fields with dropdown menus for selecting source and target objects. The fields include:

- Mobile Phone ( C\_MobilePhone )
- Contact Phone ( PrimaryPhoneNumber )
- Advanced Mode ( SourceCampaignId )
- SC Custom Number 1 ( SCCustomNumber1\_c )
- Advanced Mode ( CX Sales Account Id )
- SC Custom Text 1 ( SCCustomText1\_c )
- 300000000762003
- Owner ID ( OwnerId )
- Advanced Mode ( Company )
- SourceCampaign.Campaign Name
- Lead Name ( Name )
- Advanced Mode ( CX Sales Contact Id )
- Primary Contact ID ( PrimaryContactId )
- Advanced Mode ( SourceCampaign.CRM Campaign Id )
- Primary Campaign ID ( PrimaryCampaignId )
- CX Sales Account Id ( C\_CX\_Sales\_Account\_Id )
- Customer ID ( CustomerId )

For this step you **must** include these mappings:

- Add Owner\_id and its value
- In Advanced mapping mode: Source.CampaignId from Eloqua campaign ID from Campaign Object > mapped into your Oracle Sales campaign number field.
- Add Account Party ID (ES Account ID) on Eloqua contact record mapped into your corresponding Oracle Sales field.

In the Update Callback Section:

- Add Lead\_id mapping to CRM campaign lead ID (Unique identifier)
- Add Oracle Sales campaign number mapping to CampaignId
- Oracle Sales Account Party Id to Eloqua LeadCampaign table's account id field

The screenshot shows a web interface titled "Update Eloqua Fields". It contains a table with two main columns: "From Oracle CX Sales" and "To Eloqua".

| From Oracle CX Sales                     | To Eloqua                                    | Unique Identifier |
|--|--|-------------------|
| Lead Identifier ( LeadId )               | CampaignLead Crm Campaign Lead Id ( Crm... ) |                   |
| SC Custom Number 1 ( SCCustomNumber1_c ) | CampaignLead Campaign Id ( CampaignId )      |                   |
| SC Custom Text 1 ( SCCustomText1_c )     | CampaignLead Crm Account Id ( CrmAccountId ) |                   |

At the bottom of the table, there is a button labeled "Add Import Field Mapping".

## Step 10: Set Conversion Flag on Lead

Updates the Lead object in Oracle Sales to set the EligibleForConversion flag to true. This requests Oracle Sales to convert the lead to an Opportunity.

- Mapping: CRM Campaign Lead ID to the LeadID field in Eloqua
- Advanced mapping: write the value 'true' to the EligibleForConversion flag field

**Note:** This process is not immediate and is asynchronous. The updates will occur after the next ESS batch job on the Oracle Sales instance.

Edit Action Cancel Save

Action Name

Action Type

Connection

Eloqua Object

CX Sales Object

**Update Oracle CX Sales Fields**

| From Eloqua  | To Oracle CX Sales   | Blank Values                                   |
|--|--|--|
| <input type="text" value="Campaign/Lead Crm Campaign Lead Id ( Crm )"/>  | <input type="text" value="Lead Identifier ( LeadId )"/>                      | <input type="text" value="Unique Identifier"/> |
| <input type="text" value="true"/>  | <input type="text" value="Eligible for Conversion ( EligibleForConverso )"/> | <input type="text" value=""/>                  |
| <input type="text" value="CX Sales Contact Id ( C_CX_Sales_Contact_ )"/> | <input type="text" value="Primary Contact ID ( PrimaryContactId )"/>         | <input type="text" value=""/>                  |

## Step 7: Update Lead Product

Updates the SalesLead object in Oracle Sales with the product record and details.

Mapping:

- CRM Campaign Lead ID into LeadID
- Advanced Mapping: into Amount
- Advanced Mapping: into Quantity
- Advanced Mapping: into Product group ID

Action Name: 7 - Update Lead Product

Action Type: Upsert

Connection: CBVE

Eloqua Object: Contact

CX Sales Object: Sales Lead

CX Sales Child Object: LeadProduct

---

Update Oracle CX Sales Fields

| From Eloqua   | To Oracle CX Sales                  |
|---|-------------------------------------|
| Crm Campaign Lead Id ( Cmcampaignleadid )                           | Lead ID ( Leadid )                  |
| Advanced Mode ⓘ Insert Object ▼<br>SourceCampaignProduct Group Id ⓘ | Product Group ID ( ProductGroupid ) |
| Advanced Mode ⓘ Insert Object ▼<br>SourceCampaignProduct amount ⓘ   | Amount ( Amount )                   |
| Advanced Mode ⓘ Insert Object ▼<br>SourceCampaignNumber of units ⓘ  | Quantity ( Quantity )               |

Add Export Field Mapping

## Step 6: Link to Existing Lead

This upsert action step will create an association from the contact to the existing Lead, by creating a LeadContact as a child object of the SalesLead.

Action Name: 8 - Link Contact to Existing Lead

Action Type: Upsert

Connection: CBVE

Eloqua Object: Contact

CX Sales Object: Sales Lead

CX Sales Child Object: LeadContact

---

Update Oracle CX Sales Fields

| From Eloqua                                   | To Oracle CX Sales           |
|---|------------------------------|
| Crm Campaign Lead Id ( Cmcampaignleadid )     | Lead ID ( Leadid )           |
| CX Sales Contact Id ( C_CX_Sales_Contact_id ) | Contact Party ID ( Partyid ) |

Add Export Field Mapping

## Step 3: Link to existing Opportunity

This upsert action step will create an association from the contact to the existing Opportunity object by creating a OpportunityContact as a child object of the Opportunity.

**Note:** Creation of this object can take up to 15 minutes.

Action Name

Action Type

Connection

Eloqua Object

CX Sales Object

CX Sales Child Object

---

Update Oracle CX Sales Fields

| From Eloqua   | To Oracle CX Sales                                     |
|---|--|
| <input type="text" value="Crm Opportunity Id ( CrmOpportunityId )"/>        | <input type="text" value="OpptyId ( OpptyId )"/>       |
| <input type="text" value="CX Sales Contact Id ( C_CX_Sales_Contact_Id1 )"/> | <input type="text" value="PERPartyId ( PERPartyId )"/> |

## Opportunity Import

**Note:** For more information on creating imports, see the [Oracle CX Sales Integration app Help Center](#).

## Filter

Filter needs to be determined to be able to fetch the data from Oracle Sales. The Filter can be a SQL query including multiple fields.

Import Name

Connection

Deleted and Merged Records ☒ Exclude ☐ Import Only Deleted Records ☐ Import Only Merged Records

CX Sales Object

Filter

Eloqua Object

For this filter to function, Opportunity ID needs to not be 'null'.


## Mapping

Update Eloqua Fields

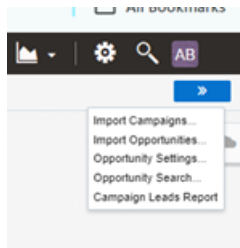
| From Oracle CX Sales  | To Eloqua  |
|---|--|
| Unique Identifier - specify the field to match an update/insert<br>Contact Email ( PrimaryContactEmailAddress ) ▼ | Email Address ( C_EmailAddress ) ▼               |
| <input type="checkbox"/> Update All Matching Records  |  |
| Contact Email ( PrimaryContactEmailAddress ) ▼  | Email Address ( C_EmailAddress )                 |
| Lead Identifier ( LeadId ) ▼  | Campaign/Lead Crm Campaign Lead Id ( CrmC ) ▼    |
| Opportunity ID ( LeadOpportunity.OptyId ) ▼   | Campaign/Lead Crm Opportunity Id ( CrmOpport ) ▼ |
| Eligible for Conversion ( EligibleForConversionF ) ▼  | Destination ( C_Destination1 ) ▼                 |
| Lead Name ( Name ) ▼  | Address 1 ( C_Address1 ) ▼                       |

## Campaign Lead data in Eloqua

For the Eloqua users of a Oracle Guided Campaigns enabled instance, a new report exists within Eloqua to see the Campaign Lead data that has been created as a result of your Lead or Opportunity qualification configuration as per the above imports and actions.

To access the report, in Eloqua, **Orchestration**  > **Campaigns** and then click **Actions** > **Campaign Leads Report**.





This gives you a new interface that shows you any Leads created by your Lead or Opportunity qualification processes. You can use the search to filter your results, for example to just a Campaign in particular.

**Tip:** You can use the search to filter your results, for example to just a Campaign in particular:

**Campaign Leads Report**

Hybrid X Search by Campaign Name

2 Items

| Campaign ID | Campaign Name                   | CRM Campaigns Lead ID | CRM Account ID  | CRM Opportunity ID |
|-------------|---------------------------------|-----------------------|-----------------|--------------------|
| 305         | Hybrid Fuel Generator Promotion | 30000054322561        | 300000542207586 |                    |
| 305         | Hybrid Fuel Generator Promotion | 300000542306584       | 300000541830558 | 300000542304856    |