



Oracle Eloqua and Salesforce

Integration Guide

Contents

- Integrating Oracle Eloqua with Salesforce** 4
- Data imports from Salesforce to Oracle Eloqua** 8
- Data exports from Oracle Eloqua to Salesforce** 12
- Steps to integrate with Salesforce** 14
- Preliminary Salesforce setup for your integration** 17
 - Setting up Salesforce user 17
 - Adding Oracle Eloqua to the list of trusted servers 19
- Preliminary Oracle Eloqua setup for your integration** 20
 - Confirming access to Oracle Eloqua integrations 20
 - Run the integration wizard for the first time 21
 - Disabling the internal and external queues 25
 - Disabling auto synchs 26
- Configuring data imports from Salesforce** 27
 - Auto synch settings for Salesforce integration 30
- Configuring data exports from Oracle Eloqua to Salesforce** 38
 - Configuring data priority for Salesforce integration 39
 - Creating Salesforce fields in Oracle Eloqua 42
 - Linking accounts to contacts for Salesforce integration 44
 - Configuring external calls to send data to Salesforce 47
 - Creating a custom campaign association external call 52
 - Testing the external calls to Salesforce 54
 - Configuring Salesforce integration programs in Oracle Eloqua 58
 - Enabling the email opt out program 59
 - Configuring the CRM update program 65
 - Choosing the right CRM update program 66

Configuring the selected SYSTEM - CRM Update program	68
Enabling Salesforce activity writing	75
Initializing and monitoring the Salesforce integration	78
Re-enabling the auto synchs	78
Enabling the internal and external queues	80
Validating auto synch processing	81
Configuring system error notifications	82
Resetting the Salesforce password in Oracle Eloqua	84

Integrating Oracle Eloqua with Salesforce

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Important: Organizations typically work with Oracle's [implementation services](#) to ensure a successful CRM integration.

This documentation provides a starting point to show how you can integrate Oracle Eloqua and Salesforce. However, your specific integration will require customizations to support your unique business needs. Our steps are based on a non-customized Oracle Eloqua instance and a non-customized Salesforce instance.

Oracle Eloqua and Salesforce integration can help you boost marketing and sales alignment and drive ROI. Connect the segmentation, campaign management, and lead generation processes in Oracle Eloqua with the lead, contact, and account management processes in Salesforce. CRM integration ensures your marketing and sales teams have accurate and detailed information about a prospect and provides a more complete picture of the buyer.

CRM integration synchronizes data between Salesforce and Oracle Eloqua:

- Synchronize account, contact, and lead data so that you can use the most accurate data in your marketing campaigns and sales engagements.

- Synchronize marketing campaign activity so marketing can provide detailed information to sales about a prospect including web activity, email opens, form submits, and more.

In addition, you can implement closed-loop reporting which enables you to attribute opportunity revenue from Salesforce to marketing campaigns. See [Closed-loop reporting with Salesforce](#) for more information on this configuration.

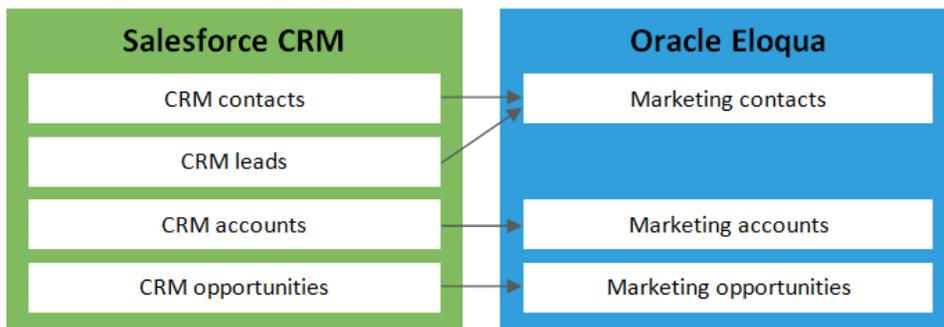
Tip: You can also use the [Salesforce Integration app](#).

About Salesforce integration with Oracle Eloqua

The CRM integration documented here uses the following process flows:

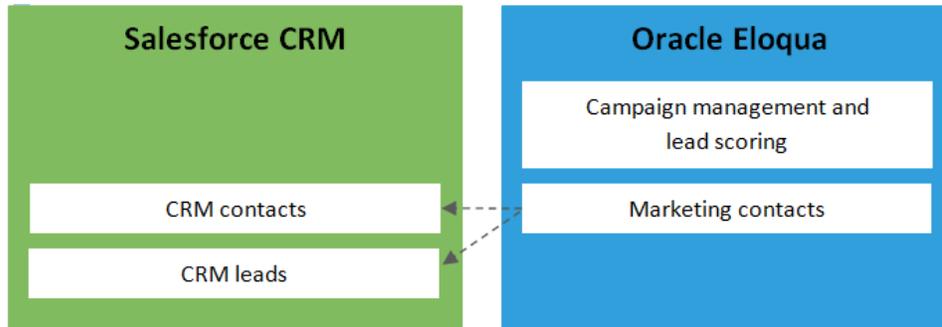
- **From Salesforce to Oracle Eloqua**

Synchronizes accounts, contacts, leads and opportunities in Salesforce with Oracle Eloqua. For more information, see [Data imports from Salesforce to Oracle Eloqua](#).



- **From Oracle Eloqua to Salesforce**

Uses contact data in Oracle Eloqua to update contacts and generate sales leads in Salesforce. A sales lead is created for each new prospect captured in Eloqua and for responses from existing contacts to a campaign or other marketing event. For more information, see [Data exports from Oracle Eloqua to Salesforce](#).



Assumptions and constraints

Before following the integration outlined in this documentation, note the following assumptions and constraints:

- Only Salesforce leads and contacts with email addresses are synchronized into Oracle Eloqua out of the box.
- Marketing activities can be logged only for known Salesforce leads and contacts at the time the activity is recorded in Oracle Eloqua.
- Oracle Eloqua does not automatically delete records. If a lead, contact, or account is deleted in CRM, the record is picked up in the *Delete* auto synch for that entity, and the corresponding Oracle Eloqua field containing that CRM ID field is to set to blank.
- Oracle Eloqua prioritizes Salesforce as the system of record, treating its data as the official, first-priority record.

Considerations

Before starting the integration process, consider the following to ensure the integration successfully meets your organization's needs:

- Make sure your data is clean before you start. For example:
 - How do you feel about the quality and completeness of your data?
 - Do you have many records with duplicate email addresses?
 - Have you cleaned up your data by running deduplication processes?
- Ensure your Salesforce administrator is involved in the integration process. Integration with Oracle Eloqua should only begin after you have a clear understanding of:
 - What business processes feed your database?
 - What business processes does your data support?
 - Do you have any data storage constraints in your CRM?
 - Are there other integrations or external databases?
 - What specific objectives is this integration intended to achieve?
- Determine how you want data to flow back into Salesforce. What leads do you want to send?

Data imports from Salesforce to Oracle Eloqua

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Oracle Eloqua imports existing account, contact, and lead data from Salesforce. After the initial import, any changed data is imported using an incremental process.

Oracle Eloqua uses auto synchs and external calls to manage the imports. When to import the data and what data to import depends on how you setup your implementation.

Note: When an account, contact, lead, or opportunity is imported to Oracle Eloqua, the Oracle Eloqua record includes the ID of the original Salesforce record.

The following table provides key details about the synchronization of different data types:

Salesforce entity	Details
Contacts	<p>Salesforce contacts are stored in the Eloqua contact database. CRM records are matched with Eloqua records using the email address.</p> <p>If a contact exists with the same email address, the contact record is updated with the latest CRM data. If there is no contact record with the same email address, a new contact</p>

Salesforce entity	Details
	<p>record is created in Oracle Eloqua. See Duplicate record processing.</p>
Leads	<p>In Salesforce, a lead is a person identified as a potential customer. Oracle Eloqua stores leads as contacts in the Oracle Eloqua contact database. CRM records are matched with Oracle Eloqua records using the email address.</p> <p>If a lead exists with the same email address, the contact record is updated with the latest CRM data. If there is no contact record with the same email address, a new contact record is created in Oracle Eloqua.</p>
Accounts	<p>Salesforce accounts are added to the Oracle Eloqua account database. CRM records are matched with the Oracle Eloqua account database using the account ID. If there is an Oracle Eloqua account with the same ID, the account record is updated with the latest CRM data. If there is no account record in Oracle Eloqua with the same ID, a new account is created.</p> <p>In Oracle Eloqua contacts are linked to accounts using the account ID. If a contact is associated with more than one account in Salesforce, only the primary account is associated with the contact in Oracle Eloqua.</p> <div data-bbox="557 1423 1424 1598" style="background-color: #e1f5fe; padding: 10px; border-radius: 5px;"> <p>Note: It is recommended that you only synch accounts from Salesforce to Oracle Eloqua.</p> </div>
Opportunities	<ul style="list-style-type: none"> • A Salesforce opportunity is not visible in Oracle Eloqua until a contact linked to the opportunity responds to an Oracle Eloqua campaign, such as by submitting a form. • A Salesforce opportunity is synchronized to Oracle Eloqua

Salesforce entity	Details
	<p>through an account that a contact is associated to. The contact to associate the opportunity with can be based on:</p> <ul style="list-style-type: none"> • The primary contact associated with the Salesforce opportunity • All contacts associated to the Salesforce opportunity • All contacts in Oracle Eloqua associated to the account that is associated to the Salesforce opportunity <ul style="list-style-type: none"> • Salesforce opportunities are matched in Oracle Eloqua using the Opportunity ID field on the opportunity object. • In Oracle Eloqua, you must define opportunity stages to match those used in Salesforce. Otherwise, the opportunity will fail to be updated in Oracle Eloqua. • In Oracle Eloqua you can search for opportunities from campaigns. To see opportunities linked to campaigns, the opportunity must be in Eloqua, the opportunity-contact link must be created via import, and the campaign response must be generated for the contact.

Duplicate record processing

Note: It is recommended that you remove duplicate sales records before integrating with Oracle Eloqua. Duplicate data history is not maintained.

Oracle Eloqua uniquely identifies records based on email address. When Salesforce contact records with the same email address are imported into Oracle Eloqua at the same time, duplicates are merged into a single record, with the highest ASCII value retained on

a field-by-field basis (Z is higher than A; 9 is higher than 1; letters are higher than numbers; lowercase is higher than uppercase).

The following table illustrates the duplicate record processing.

Email	First name	Last Name	Company	Address	Job Title
Salesforce contact records					
bob@example.com	Bob	Smith	Example Inc.	401 Island Parkway Redwood Shores, CA	Marketing Director
bob@example.com	Robert	Smith	Example Ltd.	104 Island Parkway Redwood Shores, CA	Director of Marketing
Oracle Eloqua contact record					
bob@example.com	Robert	Smith	Example Ltd.	401 Island Parkway Redwood Shores, CA	Marketing Director

Data exports from Oracle Eloqua to Salesforce

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

To update Salesforce, the integration uses Program Builder and external calls. Oracle Eloqua data is sent to Salesforce whenever an event occurs in Oracle Eloqua. For example, when a contact submits an Oracle Eloqua form (the event), form processing rules add the contact the update program. Oracle Eloqua always updates contacts in Salesforce if they exist. Oracle Eloqua creates leads based on your implementation setup. The data sent over depends on how you setup your implementation.

Oracle Eloqua maintains the relationship between Salesforce records using the Salesforce record identifier (CRM ID). The CRM ID of each record imported from Salesforce is retained on the corresponding record created or updated in Oracle Eloqua.

This table provides some key details about the synchronization of the different data types:

Oracle Eloqua data	Details
Lead	A lead is created in Salesforce when new contacts are added to the CRM Update program through events such as form submits, list uploads, and marketing campaigns.

Note: Lead assignment is managed strictly in Salesforce

Oracle Eloqua data	Details
	<p>and can be triggered upon lead creation from Oracle Eloqua.</p>
Contact	<p>Oracle Eloqua sends contact data to Salesforce when the contact record has a Salesforce Contact ID. Oracle Eloqua will not create new contacts in Salesforce.</p> <p>If the contact record does not have a Salesforce Contact ID, Oracle Eloqua creates a new lead in Salesforce.</p>
Marketing activities	<p>Specific Oracle Eloqua-generated activities, such as email opens and email clickthroughs, can be exported to a Salesforce Activity Object as a completed task for known Salesforce leads and contacts.</p> <p>You can integrate campaign and response data by implementing closed-loop reporting. See Closed-loop reporting with Salesforce for more information.</p>
Account	<p>Accounts in Salesforce are not updated when data is synchronized from Oracle Eloqua.</p>

Steps to integrate with Salesforce

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

The following table provides an overview of the steps you need to complete to integrate Salesforce and Oracle Eloqua:

Step	Description
Preliminary Salesforce setup	Complete the initial setup in Salesforce which involves creating a CRM integration user and adding Oracle Eloqua to the list of trusted servers. See Preliminary Salesforce setup for your integration .
Preliminary Oracle Eloqua setup	Complete the initial setup in Oracle Eloqua. This involves the following: <ul style="list-style-type: none">• Confirming access to Salesforce using the CRM integration user• Running the CRM Integration Wizard to start the integration process• Disabling queues and auto synchs so that the systems do not synchronize while you complete the configuration See Preliminary Oracle Eloqua setup for your integration .
Configure data	Configure the auto synchs to import Salesforce account,

Step	Description
synchronization from Salesforce to Oracle Eloqua	<p>contact, and lead data to Oracle Eloqua.</p> <p>See Configuring data imports from Salesforce.</p>
Configure data synchronization between Oracle Eloqua and Salesforce	<p>Configure how to synch data from Oracle Eloqua to Salesforce. This configuration involves the following:</p> <ul style="list-style-type: none"> • Configuring which data sources have priority to update Oracle Eloqua data • Setup any contact and account record fields in Oracle Eloqua that are needed for data synchronization • Specifying how to link accounts to contacts • Configuring external calls that will create and update in Salesforce • Creating custom web links for Salesforce users • Enabling the CRM Email Opt Out program which ensures email opt-out settings are synchronized • Configuring the program used to trigger updates to leads and contacts in Salesforce. • Enabling activity writing so that Oracle Eloqua-tracked activities, such as email clickthroughs, website visits, and form submissions, can be written to Salesforce as closed tasks <p>See Configuring data exports from Oracle Eloqua to Salesforce.</p>
Initializing and monitoring the Salesforce integration	<p>Re-initialize the systems that were disabled during the preliminary setup, then configure the notifications needed to</p>

Step	Description
	<p data-bbox="589 268 1349 363">monitor and maintain the integration. This involves the following:</p> <ul data-bbox="643 415 1219 594" style="list-style-type: none"><li data-bbox="643 415 980 447">• Enabling the auto synchs<li data-bbox="643 489 1182 520">• Enabling the internal and external queues<li data-bbox="643 562 1219 594">• Configuring system notifications about errors <p data-bbox="589 657 1390 688">See Initializing and monitoring the Salesforce integration.</p>

Preliminary Salesforce setup for your integration

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Before you begin the Salesforce integration, complete these preliminary tasks in Salesforce:

- [Setting up Salesforce user](#)
- [Adding Oracle Eloqua to the list of trusted servers](#)

Note: To complete the second task, you must identify Oracle Eloqua by IP. If you do not have the Oracle Eloqua IP ranges, contact [My Oracle Support](#) (<https://support.oracle.com>).

Setting up Salesforce user

It is strongly recommended that you create a unique Salesforce user exclusively for the ongoing data exchange between Oracle Eloqua and Salesforce. This can help properly attribute marketing contribution to the capturing and updating of Salesforce records (leads and contacts). It can also help enhance some of the Salesforce reporting capabilities with Oracle Eloqua data. As a result, this can support more accurate

troubleshooting, allowing you to distinguish changes made by another user for the integration from changes made for other reasons.

To create the Salesforce user for integration:

1. Create a new Salesforce user. We recommend using the following settings to make the user easily identifiable:

- **First Name** - Eloqua
- **Last Name** - Marketing
- **Alias** - Eloqua
- **Email** - your email address
- **Username** - eloqua@example.com where *example* is your organization domain

 **Note:** Ideally, the password for this Salesforce user does not expire. Although password changes can be made within Oracle Eloqua, it requires that your Customer Administrator manages regular updates within Oracle Eloqua. See [Resetting the Salesforce password in Oracle Eloqua](#)

2. Configure the other fields as per your requirements.

3. Set up the user with sufficient access to create and update Salesforce data. We recommend the following:

- Administrative access (recommended)
- Access to:
 - Create, update, and read Salesforce leads
 - Update and read Salesforce contacts
 - Read Salesforce accounts
 - Create campaign members and update campaign member status

- Create and read campaigns
- Create tasks on contact and lead records

4. Record this user name and password as needed.

Adding Oracle Eloqua to the list of trusted servers

You must permit Oracle Eloqua servers to interact with Salesforce. To do this, add the Oracle Eloqua IP addresses to the Salesforce list of approved or trusted servers (allowlist). For a list of IP ranges to add, see this [knowledge base article](#) or contact [My Oracle Support](#) (<https://support.oracle.com>).

Note: If you want to allow Oracle Eloqua employees (including support) to access your CRM system and troubleshoot issues, please [complete this form](#). After you submit the form, you will receive Oracle Eloqua's corporate IPs to add to your allowlist.

Preliminary Oracle Eloqua setup for your integration

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Before you begin the Salesforce integration, complete these preliminary tasks in Oracle Eloqua:

- [Confirming access to Oracle Eloqua integrations](#)
- [Run the integration wizard for the first time](#)
- [Disabling the internal and external queues](#)
- [Disabling auto synchs](#)

Note: Before you complete the preliminary integration setup in Oracle Eloqua, complete the tasks in [Preliminary Salesforce setup for your integration](#). You must be a member of the Customer Administrator security group in Oracle Eloqua to access the integration functionality and perform the configuration tasks.

Confirming access to Oracle Eloqua integrations

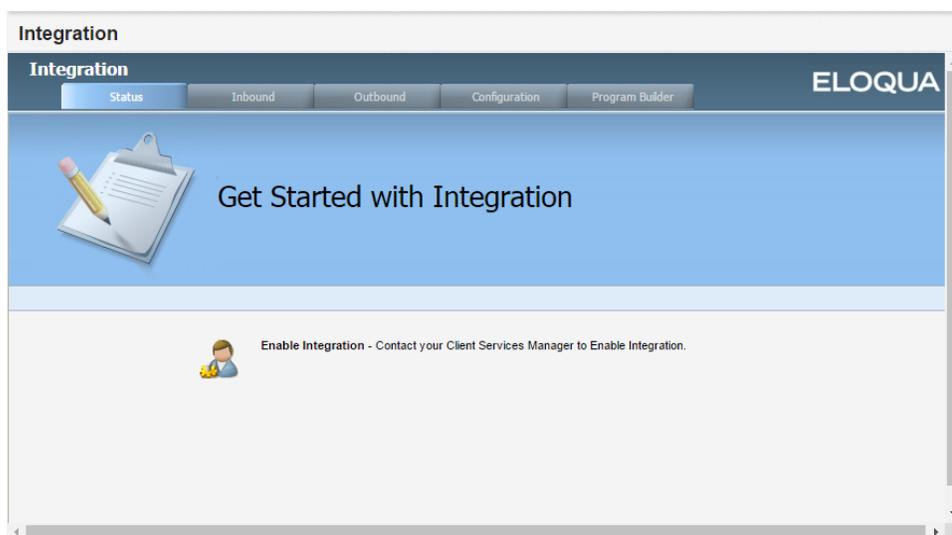
You must confirm that you have access to Oracle Eloqua's integration functionality. If you discover integration is not enabled in your Oracle Eloqua instance, please log in to [My](#)

Oracle Support (<https://support.oracle.com>) and create a service request.

To confirm access to the integration area in Oracle Eloqua

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.

If the feature is not available, the screen indicates that you have to contact support.



Run the integration wizard for the first time

The CRM Integration Setup Wizard helps start the integration process. The wizard will verify the connection to Salesforce and setup the default configuration. Later, you disable some of these default configurations so that you can customize the integration with Salesforce.

Note: The CRM Integration Setup Wizard only needs to be run one time.

Before you begin:

- [Confirm that you have access to Oracle Eloqua's integration functionality](#)
- Complete the preliminary setup in Salesforce. See [Preliminary Salesforce setup for your integration](#) for more information.

To run the integration wizard:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*. If you have not run the CRM Integration Setup Wizard yet, the *CRM Integration Setup* window is displayed.
3. Click the **Configuration** tab.
4. Enter the user name and password for the Salesforce user [created for the integration](#). This is the user that Oracle Eloqua will use to connect with Salesforce.
5. To use a testing environment for the integration instead of your production instance, select the **Use Sandbox** check box. After you complete the integration, reset the user to your production environment and resend the data.

 **Note:** If you are using the Enterprise trim of Oracle Eloqua, you can request a sandbox. If you have the Basic or Standard trims, this is an add-on function. For more information on how to request access to the Replication Sandbox, please contact your account representative.

6. Enter the email address for the administrator who should be informed of critical errors in the **Notification Email** field. You can enter multiple email addresses by separating them with a semi-colon (;).

CRM Integration Setup

Salesforce

[Setup Overview](#)

This area is used to configure Eloqua with your CRM system - marketing activities can be recorded and made visible to your sales people in your CRM. You will have a basic integration that will allow you to send an email batch, record the email opens, clickthroughs and website visits of your prospects into your CRM database.

[CRM Account Information](#)

Enter a CRM Username and password that Eloqua will use to login to your CRM system and write marketing activities to your CRM and update or create contact and lead records.

CRM User Name

CRM Password

Confirm Password

Use Sandbox

[Error Notification](#)

Eloqua can send a notification to a particular email address if a critical error occurs while communicating with your CRM system.

Notification Email



7. Click **Start CRM Integration**.

The wizard sets up a standard CRM integration with Salesforce. Oracle Eloqua displays the progress of the setup. The following table provides details about each setup step.

Setup step	Details
Validate CRM login	Checks that the CRM user name and password are accurate and that Oracle Eloqua can communicate with the Salesforce system.
Setup Eloqua business logic	<p>Sets up the integration logic in Oracle Eloqua, which dictates the following:</p> <ul style="list-style-type: none"> • When a lead in Salesforce doesn't exist, Oracle Eloqua creates a new lead in Salesforce. • When a lead in Salesforce exists, Oracle Eloqua updates the lead in Salesforce. • When a contact in Salesforce exists, Oracle Eloqua updates the contact in Salesforce. It is recommended that you do not create contacts or accounts from Oracle Eloqua.

Setup step	Details
	<p>Note: This logic will be updated and customized using automated programs in Program Builder.</p>
<p>Examine CRM system configuration and field names</p>	<p>Pulls all of the data objects and associated fields from Salesforce for integration purposes.</p> <p>Note: The wizard attempts to create the necessary external calls to set up your integration and presents errors for fields that are not accessible. Ensure that the specified fields are accessible to the Salesforce user as readable and writable fields. You cannot re-run the wizard, but you can manually update the external calls to select and map the fields. If these fields are not part of your set of integrated fields, then you can ignore this error.</p>
<p>Setup Eloqua data push and pull settings</p>	<p>Sets up external calls in Oracle Eloqua to update Salesforce and synchronize data back from Salesforce to Oracle Eloqua.</p>
<p>Setup Automatic Data Syncs from CRM system</p>	<p>Sets up the auto syncs to pull data from Salesforce.</p>
<p>Create web links</p>	<p>Creates the Oracle Eloqua web links <i>Contact Activity Overview</i> and <i>Lead Activity Overview</i>, which can be added to page layouts in Salesforce. This step also detects the current custom links on the lead and contact entities within Salesforce.</p>
<p>Enable CRM integration</p>	<p>Completes the CRM integration process and enables internal events and external calls.</p>

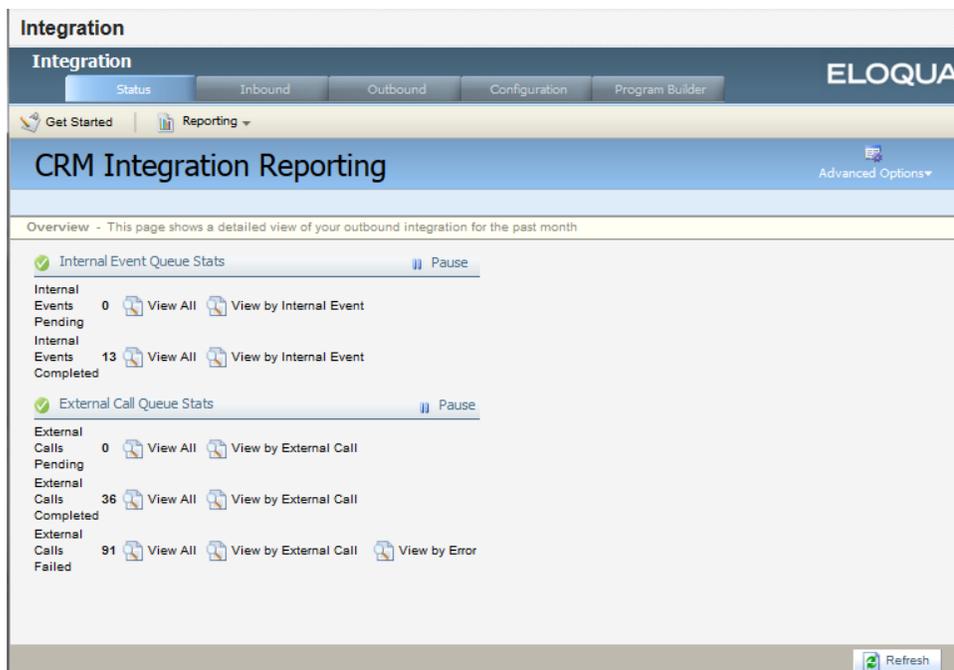
After you finish: Disable the internal and external queues and auto syncs.

Disabling the internal and external queues

The CRM Integration Setup Wizard set up the processes to support the transfer of data between Oracle Eloqua and Salesforce. You must disable these processes so that you can customize the integration. Disabling these processes stops importing and exporting data during the configuration.

To disable the internal and external queues:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Status** tab. Click **Reporting**, then click **Integration Reporting**. The statuses of the internal and external event queues are displayed.



If *Pause* is displayed for either of the queues, the data transfer is active.

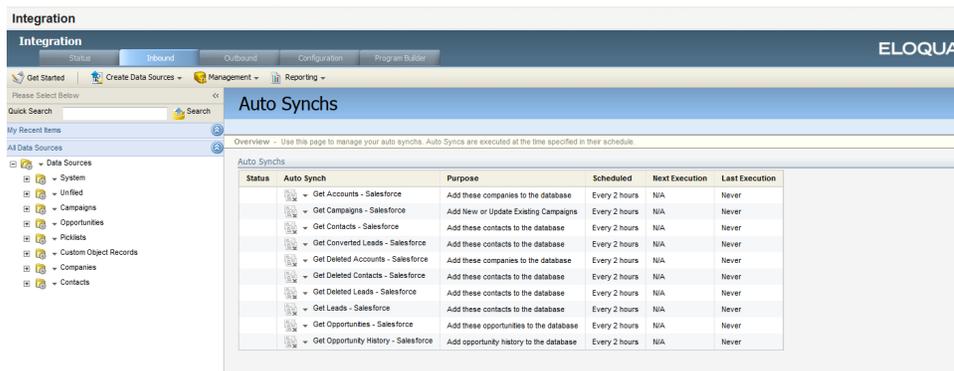
4. Click **Pause**.

Disabling auto synchs

The CRM Integration Setup Wizard set up the processes to import data from Salesforce. You must disable the auto synchs to stop the imports during the configuration. For more information about auto-synchs, see [CRM integration: auto-synchs](#).

To disable auto synchs:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Inbound** tab. Click **Management** and then click **Auto Synchs**.



The screenshot shows the ELOQUA Integration Management console. The 'Auto Synchs' page is active, displaying a table of auto synchs. The table has the following columns: Status, Auto Synch, Purpose, Scheduled, Next Execution, and Last Execution. The table contains 10 rows of auto synchs, all with a status of 'On' and a scheduled frequency of 'Every 2 hours'.

Status	Auto Synch	Purpose	Scheduled	Next Execution	Last Execution
On	Get Accounts - Salesforce	Add these companies to the database	Every 2 hours	N/A	Never
On	Get Campaigns - Salesforce	Add New or Update Existing Campaigns	Every 2 hours	N/A	Never
On	Get Contacts - Salesforce	Add these contacts to the database	Every 2 hours	N/A	Never
On	Get Converted Leads - Salesforce	Add these contacts to the database	Every 2 hours	N/A	Never
On	Get Deleted Accounts - Salesforce	Add these companies to the database	Every 2 hours	N/A	Never
On	Get Deleted Contacts - Salesforce	Add these contacts to the database	Every 2 hours	N/A	Never
On	Get Deleted Leads - Salesforce	Add these contacts to the database	Every 2 hours	N/A	Never
On	Get Leads - Salesforce	Add these contacts to the database	Every 2 hours	N/A	Never
On	Get Opportunities - Salesforce	Add these opportunities to the database	Every 2 hours	N/A	Never
On	Get Opportunity History - Salesforce	Add opportunity history to the database	Every 2 hours	N/A	Never

4. For each auto synch in the list, select **Disable Auto Synch** from the drop-down list.

Configuring data imports from Salesforce

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Note: You must be a member of the Customer Administrator security group in Oracle Eloqua to access the integration functionality and perform the configuration tasks.

To integrate Salesforce and Oracle Eloqua, you must configure auto synchs. Auto synchs are scheduled imports to Oracle Eloqua. An auto synch defines the following:

- the import schedule
- the fields to synchronize
- the actions that Oracle Eloqua performs when the data is imported

After you ran the [CRM Integration Setup Wizard for the first time](#), Oracle Eloqua set up the following auto synchs for importing account, contact, and lead data:

- Get Accounts
- Get Contacts
- Get Leads
- Get Converted Leads
- Get Deleted Accounts

- Get Deleted Contacts
- Get Deleted Leads

For initial setup, you [disabled all the auto synchs](#) until you configured them. In your environment, you may have additional auto synchs for campaigns or opportunities. The configuration below is specific to importing Salesforce leads, contacts, and accounts.

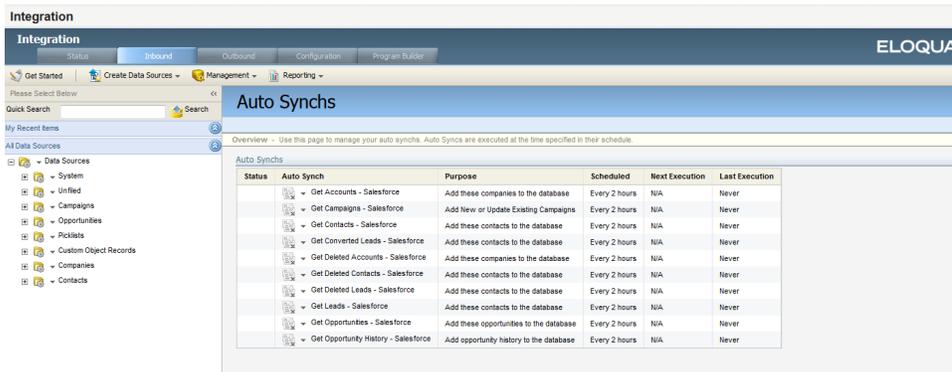
Before you begin:

- If you haven't already, complete the [preliminary Salesforce and Oracle Eloqua setup](#).
- Each auto synch configuration determines what fields are synchronized from Salesforce and how often they are synchronized. Review the settings in your auto synchs and customize them to meet your requirements if necessary.
- Refer to [Auto synch settings for Salesforce integration](#) for the recommended settings for each auto synch.

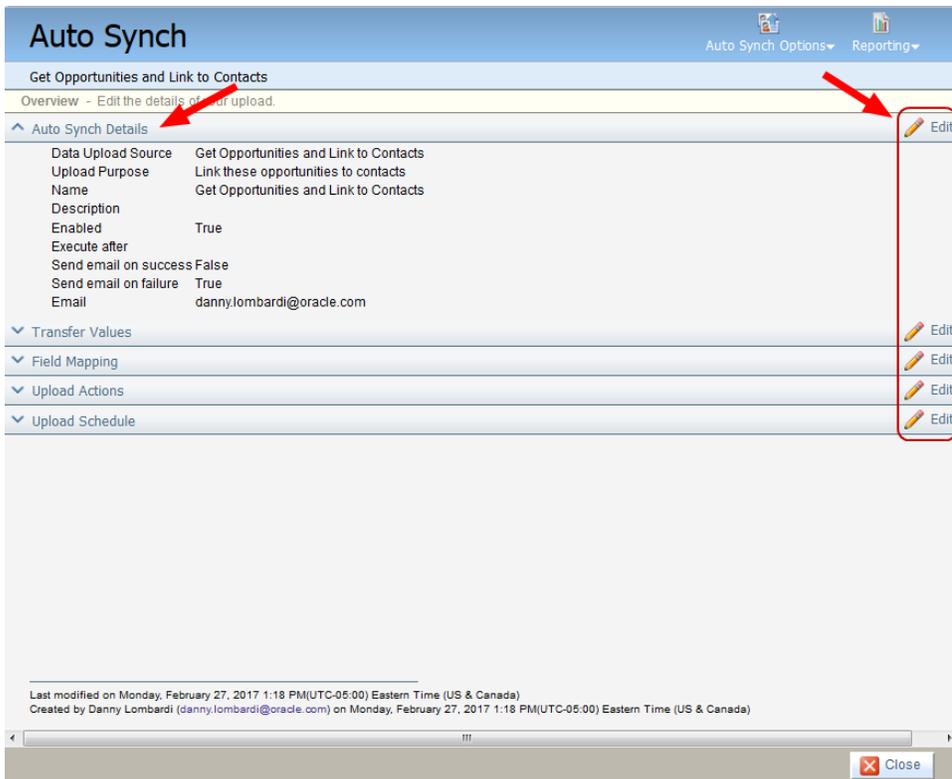
To configure auto synchs:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Inbound** tab.

4. Click **Management > Auto Synchs**. All the existing auto synchs are listed.



5. For each auto synch in the list, click **Auto Synch Details** from the drop-down.
6. Click **Edit** and configure the auto synch. Refer to [Auto synch settings for Salesforce integration](#) for the recommended settings for account, contact, and lead auto synchs.



The following auto synchs need to be configured with the specifics of your field mappings between Oracle Eloqua and Salesforce fields:

- Get Accounts
- Get Contacts
- Get Leads

The following auto synchs may require modifications to the filter criteria, depending on how often you want these synchronizations to run:

- Get Deleted Accounts
- Get Deleted Contacts
- Get Deleted Leads

Auto synch settings for Salesforce integration

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

After you ran the [CRM Integration Setup Wizard for the first time](#), Oracle Eloqua set up the following auto synchs for importing account, contact, and lead data:

- Get Accounts
- Get Contacts
- Get Leads
- Get Converted Leads
- Get Deleted Accounts

- Get Deleted Contacts
- Get Deleted Leads

When configuring auto synchs for Salesforce integration, use the settings below as guidelines. If a specific setting is not mentioned, you can use the existing configuration or leave it blank.

For information about setting up auto synchs for Salesforce, see [Configuring data imports from Salesforce](#).

Upload schedule

You can change the auto synch synchronization schedule for account and contact data as required. It is recommended that you specify the following values:

- *Auto Synch Runs on Selected Days at 10pm EST?* – Select **No**.
- *Auto-Synch Runs or Repeats Once Every 2.0 hours.* You can specify values ranging from 30 minutes to 24 hours.
- *Auto Synch proceeds by Standard Business Schedule?* – Select **No**.
- In the calendar, specify the day the auto synch is to run, the start time to run the auto synch, and the end time.

Get Accounts auto synch

Auto Synch Setting Group	Setting
Auto Synch Details	Upload Purpose: Add these companies to the database.

Auto Synch Setting Group	Setting
Transfer Values	Action: Retrieve Entity: Account Filter Details: Last Modified Date/Greater Than or Equal/Last Successful Upload
Field Mapping	Uniquely Match On: Eloqua Company Field: SFDC Account ID Perform a case-sensitive match: True
Upload Actions	Add to Company Group: SYSTEM - SFDC Accounts

Get Contacts auto synch

Auto Synch Setting Group	Setting
Auto Synch Details	Upload Purpose: Add these contacts to the database.
Transfer Values	Action: Retrieve Entity: Account Filter Details: Last Modified Date/Greater Than or Equal/Last Successful Upload AND Email/Equals/%@%
Field Mapping	Uniquely Match On: Eloqua Company Field: Email Address Perform a case-sensitive match: False
Upload Actions	Add to Contact Group: SYSTEM - SFDC Contacts

Get Leads auto synch

Auto Synch Setting Group	Setting
Auto Synch Details	Upload Purpose: Add these contacts to the database.
Transfer Values	Action: Retrieve Entity: Lead Filter Details: Last Modified Date/Greater Than or Equal/Last Successful Upload AND Email/Equals/%@% AND Converted/Equals/FALSE
Field Mapping	Uniquely Match On: Eloqua Contact Field: Email Address Perform a case-sensitive match: False
Upload Actions	Add to Contact Group: SYSTEM - SFDC Leads

Get Deleted Accounts auto synch

Auto Synch Setting Group	Setting
Auto Synch Details	Upload Purpose: Add these companies to the database.

Auto Synch Setting Group	Setting														
Transfer Values	Action: Get Deleted Entity: Account Filter Details: Create the filter based on your field mappings.														
Field Mapping	Uniquely Match On: Eloqua Company Field: SFDC Account ID Perform a case-sensitive match: True Advanced Options: <table border="0"> <tr> <td>Account ID</td> <td>></td> <td>SFDC Account ID</td> <td>False</td> <td>False</td> </tr> <tr> <td>Deleted Date</td> <td>></td> <td>[blank]</td> <td>False</td> <td>False</td> </tr> </table> Update records with a set value: <table border="0"> <tr> <td>Field</td> <td>Set to value</td> </tr> <tr> <td>SFDC Account ID</td> <td>[blank]</td> </tr> </table>	Account ID	>	SFDC Account ID	False	False	Deleted Date	>	[blank]	False	False	Field	Set to value	SFDC Account ID	[blank]
Account ID	>	SFDC Account ID	False	False											
Deleted Date	>	[blank]	False	False											
Field	Set to value														
SFDC Account ID	[blank]														
Upload Actions	Add to Contact Group: SYSTEM - SFDC Deleted Accounts														

Get Converted Leads auto synch

Auto Synch Setting Group	Setting
Auto Synch Details	Upload Purpose: Add these contacts to the database.
Transfer Values	Action: Retrieve Entity: Lead

Auto Synch Setting Group	Setting												
	<p>Filter Details:</p> <p>Last Modified Date/Greater Than or Equal/Last Successful Upload</p> <p>AND</p> <p>Email/Equals/%@%</p> <p>AND</p> <p>Converted/Equals/TRUE</p>												
Field Mapping	<p>Uniquely Match On: Eloqua Contact Field: SFDC Lead ID</p> <p>Perform a case-sensitive match:</p> <p>True</p> <p>Advanced Options:</p> <table border="1" data-bbox="440 1140 1256 1266"> <tbody> <tr> <td>Lead: Email</td> <td>> Email Address</td> <td>False</td> <td>False</td> </tr> <tr> <td>Lead: Lead ID</td> <td>> SFDC LeadID</td> <td>False</td> <td>False</td> </tr> </tbody> </table> <p>Update records with a set value:</p> <table border="1" data-bbox="440 1440 1256 1528"> <thead> <tr> <th>Field</th> <th>Set to value</th> </tr> </thead> <tbody> <tr> <td>SFDC LeadID</td> <td>[blank]</td> </tr> </tbody> </table>	Lead: Email	> Email Address	False	False	Lead: Lead ID	> SFDC LeadID	False	False	Field	Set to value	SFDC LeadID	[blank]
Lead: Email	> Email Address	False	False										
Lead: Lead ID	> SFDC LeadID	False	False										
Field	Set to value												
SFDC LeadID	[blank]												
Upload Actions	Add to Contact Group: SYSTEM - SFDC Converted Leads												

Get Deleted Contacts auto synch

Auto Synch Setting Group	Setting																
Auto Synch Details	Upload Purpose: Add these contacts to the database.																
Transfer Values	Action: Get Deleted Entity: Contact Filter Details: Create the filter based on your field mappings.																
Field Mapping	Uniquely Match On: Eloqua Contact Field: SFDC Contact ID Perform a case-sensitive match: True Advanced Options: <table border="0" style="margin-left: 20px;"> <tr> <td>Contact ID</td> <td>></td> <td>SFDC ContactID</td> <td>False</td> <td>False</td> </tr> <tr> <td>Deleted Date</td> <td>></td> <td>[blank]</td> <td>False</td> <td>False</td> </tr> </table> Update records with a set value: <table border="0" style="margin-left: 20px;"> <thead> <tr> <th>Field</th> <th>Set to value</th> </tr> </thead> <tbody> <tr> <td>SFDC ContactID</td> <td>[blank]</td> </tr> <tr> <td>SFDC AccountID</td> <td>[blank]</td> </tr> </tbody> </table>	Contact ID	>	SFDC ContactID	False	False	Deleted Date	>	[blank]	False	False	Field	Set to value	SFDC ContactID	[blank]	SFDC AccountID	[blank]
Contact ID	>	SFDC ContactID	False	False													
Deleted Date	>	[blank]	False	False													
Field	Set to value																
SFDC ContactID	[blank]																
SFDC AccountID	[blank]																
Upload Actions	Add to Contact Group: SYSTEM - SFDC Deleted Contacts																

Get Deleted Leads auto synch

Auto Synch Setting Group	Setting														
Auto Synch Details	<p>Upload Purpose: Add these contacts to the database.</p>														
Transfer Values	<p>Action: Get Deleted</p> <p>Entity: Lead</p> <p>Filter Details: Create the filter based on your field mappings.</p>														
Field Mapping	<p>Uniquely Match On: Eloqua Contact Field: ZZ - SFDCLeadID</p> <p>Perform a case-sensitive match: True</p> <p>Advanced Options:</p> <table border="0"> <tr> <td>Lead ID</td> <td>></td> <td>ZZ - SFDCLead ID</td> <td>False</td> <td>False</td> </tr> <tr> <td>Deleted Date</td> <td>></td> <td>[blank]</td> <td>False</td> <td>False</td> </tr> </table> <p>Update records with a set value:</p> <table border="0"> <thead> <tr> <th>Field</th> <th>Set to value</th> </tr> </thead> <tbody> <tr> <td>ZZ - SFDCLeadID</td> <td>[blank]</td> </tr> </tbody> </table>	Lead ID	>	ZZ - SFDCLead ID	False	False	Deleted Date	>	[blank]	False	False	Field	Set to value	ZZ - SFDCLeadID	[blank]
Lead ID	>	ZZ - SFDCLead ID	False	False											
Deleted Date	>	[blank]	False	False											
Field	Set to value														
ZZ - SFDCLeadID	[blank]														
Upload Actions	Add to Contact Group: SYSTEM - SFDC Deleted Leads														

Configuring data exports from Oracle Eloqua to Salesforce

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

After you configured the imports from Salesforce to Oracle Eloqua, you must configure the export of data from Oracle Eloqua to Salesforce.

Configuring data exports involves the following tasks:

- [Configuring data priority for Salesforce integration](#)
- [Creating Salesforce fields in Oracle Eloqua](#)
- [Linking accounts to contacts for Salesforce integration](#)
- [Configuring external calls to send data to Salesforce](#)
- [Configuring Salesforce integration programs in Oracle Eloqua](#)
- [Enabling Salesforce activity writing](#)

Note: The folders and paths discussed in this document are the defaults or recommendations. Your environment may use different names.

Configuring data priority for Salesforce integration

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Data priority specifies the order in which Oracle Eloqua evaluates sources of contact and account data and determines whether to update the data in the Oracle Eloqua database.

With a Salesforce integration, we recommend prioritizing CRM data imports over other data sources such as list uploads. This assumes that your CRM data is current and accurate and that your Salesforce contacts are given higher priority than your CRM leads.

The recommended data priority order is:

1. Bulk API
2. Get CRM Accounts
3. Get CRM Contacts
4. Get CRM Leads

These data priorities will be shared by multiple data sources. For example, the data sources Get CRM Contacts and Delete CRM Contacts will use the same data priority setting of Get CRM Contacts. This ensures that Oracle Eloqua omits the specified ID values and also allows for the ID values to be repopulated should the deleted entity be recreated in Salesforce.

To configure data import priority and the data sources:

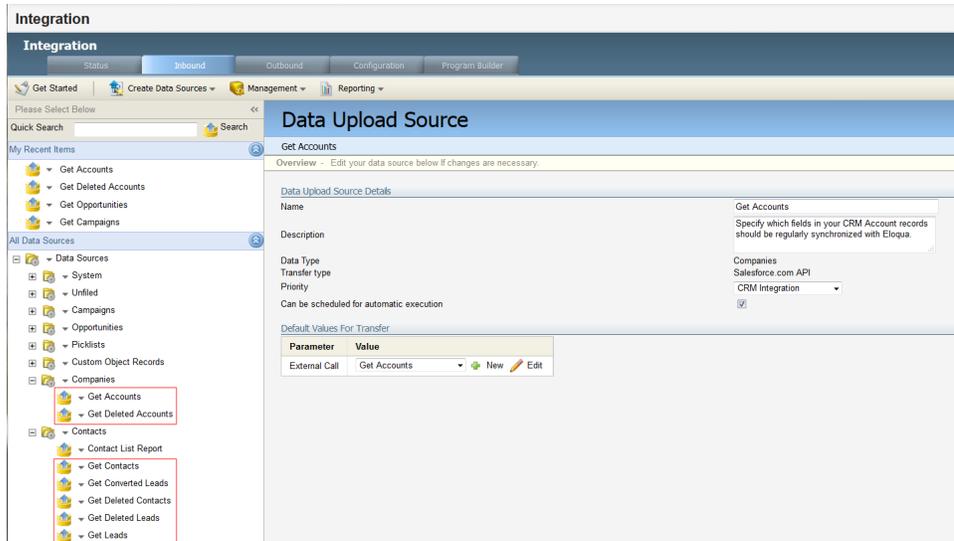
1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Inbound** tab.
4. From the *Management* menu, click **Data Priority Order**.
5. Create the following list of data import priorities.

The order of the data priorities must be the order listed below.

Data Import Priorities	
Name	
	Bulk API
	Get CRM Accounts
	Get CRM Contacts
	Get CRM Leads

 **Tip:** After the running the CRM Integration Wizard, Oracle Eloqua created a default list of data import priorities. You can rename those items to match the list above or remove them and create your own list.

6. In the *Inbound* tab, expand the folders under **All Data Sources**. This folder lists all the data sources corresponding to each [auto synch set up earlier](#).



7. For each data source listed, set the priority using the **Priority** drop-down list with the following settings:

Data source	Data priority
Get Accounts	Get CRM Accounts
Get Deleted Accounts	Get CRM Accounts
Get Contacts	Get CRM Contacts
Get Deleted Contacts	Get CRM Contacts
Get Leads	Get CRM Leads
Get Deleted Leads	Get CRM Leads
Get Converted Leads	Get CRM Leads

Creating Salesforce fields in Oracle Eloqua

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

By default, Oracle Eloqua offers a number of [commonly used contact fields](#) and [account fields](#). But to complete the integration with Salesforce, you might need to create additional fields to match those in your Salesforce.

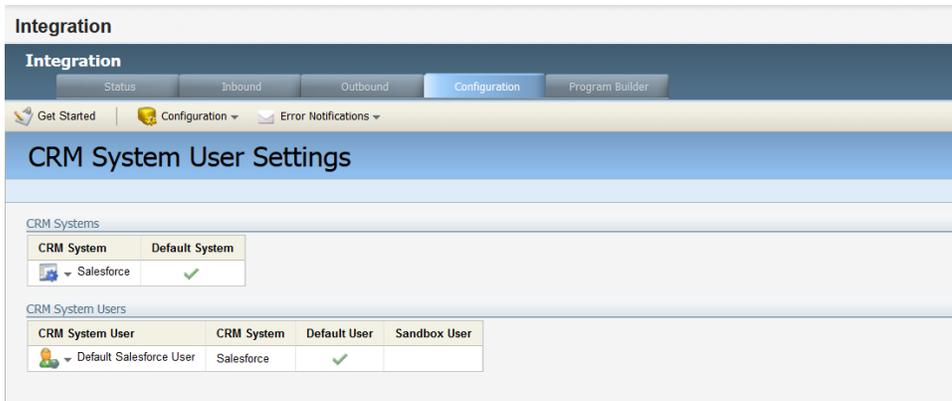
To do this, you can clone the Salesforce fields that you need. When you clone, any picklists associated to a field are also created.

Before you begin:

- Oracle Eloqua does not prevent you from creating duplicate fields. Before you create a field, confirm that it does not already exist to avoid creating duplicate fields. To view the contact and account fields available in Oracle Eloqua, click **Settings** , then click **Manage Fields & Views**. See [Fields and views](#) for more information.
- Both Salesforce lead and contact data is stored in the Oracle Eloqua contact database. It is not necessary to create a contact field for each Salesforce entity type.
- It is recommended that you verify that picklist values are consistent across leads and contacts prior to record synchronization.

To create required Salesforce fields in Oracle Eloqua:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Configuration** tab.
4. From the **Configuration** menu, click **Manage CRM System Users**.



5. Click  next to Salesforce, then click **List Fields**.
6. In the **CRM System Fields** window, select the entity for which you would like to view fields.

To view this	Select the following entity
Salesforce Accounts	Account (account)
Salesforce Contacts	Contact (contact)
Salesforce Leads	Lead (lead)

The list of Salesforce entity fields appears.

7. Select the fields you want to create in Oracle Eloqua and click **Add Selected Fields**.
 - You can add Salesforce fields to Oracle Eloqua one at a time or in a batch.
 - If a field has an associated picklist, you *must create each one individually*, in order to correctly create

the associated picklists.

CRM System Fields

Salesforce

Entity

Entity: Lead (lead)

Fields

	Display Name	Internal Name	Field Type	Field Length	Creatable	Updateable	Add
	Address	Address	address	0	✗	✗	<input type="checkbox"/>
	Annual Revenue	AnnualRevenue	currency	0	✓	✓	<input checked="" type="checkbox"/>
	City	City	string	40	✓	✓	<input type="checkbox"/>
	Clean Status	CleanStatus	picklist	40	✓	✓	<input type="checkbox"/>
	Create Contact Field	Company	string	255	✓	✓	<input checked="" type="checkbox"/>
	Copy Picklist	CompanyDunsNumber	string	9	✓	✓	<input type="checkbox"/>
	Converted	IsConverted	boolean	0	✓	✗	<input checked="" type="checkbox"/>
	Converted Account ID	ConvertedAccountId	reference	18	✗	✗	<input type="checkbox"/>
	Converted Contact ID	ConvertedContactId	reference	18	✗	✗	<input type="checkbox"/>
	Converted Date	ConvertedDate	date	0	✗	✗	<input type="checkbox"/>
	Converted Opportunity ID	ConvertedOpportunityId	reference	18	✗	✗	<input type="checkbox"/>
	Country	Country	string	80	✓	✓	<input type="checkbox"/>
	Created By ID	CreatedById	reference	18	✗	✗	<input type="checkbox"/>
	Created Date	CreatedDate	datetime	0	✗	✗	<input type="checkbox"/>
	Current Generator(s)	CurrentGenerators__c	string	100	✓	✓	<input type="checkbox"/>
	D&B Company ID	DandbCompanyId	reference	18	✓	✓	<input type="checkbox"/>
	Data.com Key	Jigsaw	string	20	✓	✓	<input type="checkbox"/>
	Deleted	IsDeleted	boolean	0	✗	✗	<input type="checkbox"/>

8. Click **Close** when you're done.

Linking accounts to contacts for Salesforce integration

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

For a Salesforce integration, you link accounts to contacts by mapping the Salesforce Account ID field. This ensures that any significant activity by a contact that is linked to an account is then associated with that account.

For more information, see [Linking accounts to contacts](#).

Before you begin:

- Complete this task after you [create Salesforce fields in Oracle Eloqua](#).
- When performing this task, ensure that you link accounts and contacts using the Salesforce Account ID field. Otherwise, contacts may be mapped to the wrong account, which could affect contact processing.

Note: After processing for the mappings has begun, it cannot be undone (rolled back or unmapped).

To link accounts to contacts:

1. Navigate to **Audience** , then **Tools**, then click **Account Linkage**.

The *Account Linkage* page opens. By default, *No Linkage* is selected unless you've already linked a contact field to an account field.

Account Linkage

Account Linkage

Alert! - Make sure that you enter the correct values in the Contact Field and Account Field, or contacts may end up being mapped to the wrong account, which will affect contact processing. Note that once processing for the mappings has begun, it cannot be undone (rolled back or unmapped).

No Linkage
 Link by mapping

2. Select **Link by mapping**.
3. Select the **SFDC Account ID** as the fields you want to use to automatically link contact and account records by. For more information, see [Linking accounts to contacts](#).

No Linkage
 Link by mapping

1) Select the fields that you would like to use to automatically link contact/account records.

Contact Field  to

Account Field 

4. Select the **Perform a case-sensitive match** check box since Salesforce uses case-sensitive matches.
5. Verify that the fields you've selected for the linkage pass the verification tests in step 2 and 3. A green check mark  signifies a successful validation.

- *Unique Account Verification:* This process checks to ensure that all accounts have a unique value for the field you've selected. If verification fails, click **Show** to see a list of duplicates, then make any adjustments as needed.
- *Check Account Dependency:* This process searches for any conflicting automated programs steps that automate account linkage through a program built in the [program builder](#) or [program canvas](#). View all of the marketing objects and processes in Oracle Eloqua that have the account field you selected as a dependency. This may include update source records, program ownership rules, deduplication or match rules, and any

automated marketing programs that use any of these rules. If the dependency check fails, navigate to the programs and remove the conflicting program steps. You must resolve all of these dependencies before you can proceed with system-level linking.

Before continuing, the page should look like this:

No Linkage
 Link by mapping

1) Select the fields that you would like to use to automatically link contact/account records.

Contact Field: to
Account Field:

Perform a case-sensitive match

2) Click the button below to verify that the accounts/companies in the system each have a unique identifier. You cannot have any company records that use the same ID in the system.

Unique Account Verification ✓

3) Before the automatic linkage can be started, we must verify that there are no 'account linkage' rules that are in place anywhere in the system. These are usually placed in a program to link account and contact records. If there are dependencies discovered - you will need to delete them to proceed. You can leave this area and return when you have removed the linkage in the system.

Check Account Dependency ✓

4) When you have completed the above steps, you can press the 'apply' button below. It will start the process of linking records, and take approximately 24 hours. After that - linkages will happen almost immediately as you add or change records in the system.

6. When both checks are successful, click **Apply** to begin processing the linkage between the contact and account fields you selected.

Configuring external calls to send data to Salesforce

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

External calls are the external CRM system calls that Oracle Eloqua uses to keep the systems synchronized. For general information about external calls, see [Managing external calls](#).

There are two types of external calls:

- **Retrieve Data:** Calls that requests data from your CRM system. These are used by auto synchs.
- **Send Data:** Calls that send data to your CRM system. These are triggered by internal events.

When sending data to Salesforce, the external call settings determine what data to send from Oracle Eloqua. External calls use a field mapping to determine which Oracle Eloqua fields to send. External calls are triggered by internal events and internal events are executed through a Program Builder program.

After running the CRM Integration Wizard, Oracle Eloqua setup the following external calls to send data for each Salesforce entity (leads, contacts, accounts):

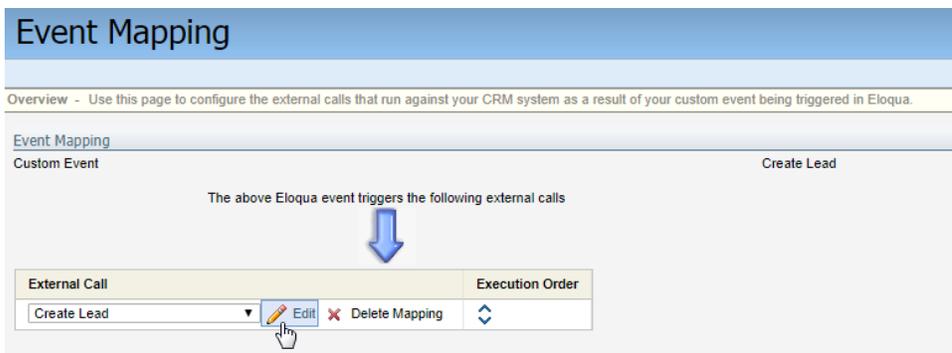
- **Create Lead:** This call is used to create a new lead in Salesforce.
- **Update Lead:** This call is used to update an existing lead in Salesforce (based on Salesforce LeadID).
- **Update Contact:** This call is used to update an existing contact in Salesforce (based on Salesforce ContactID).
- **Associate Lead with Campaign:** This call is used to associate campaign members (based on Salesforce LeadID) with a Salesforce campaign. If you are implementing closed-loop reporting, see [Closed-loop reporting with Salesforce](#) for more information on setting up these external calls for campaign associations.
- **Associate Contact with Campaign:** This call is used to associate campaign members (based on Salesforce ContactID) with a Salesforce campaign. It also updates the status of the campaign response. If you are implementing closed-loop reporting, see [Closed-loop reporting with Salesforce](#) for more information on setting up these external calls for campaign associations.

Before you begin:

- Create the required [Salesforce fields in Oracle Eloqua](#)
- Setup how [accounts are linked to contacts](#)

To configure outbound external calls for your Salesforce integration:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Outbound** tab.
4. In the left panel, expand the **Custom** folder under the *Internal Events* tab, then expand **Custom Contact Events**.
5. Click on name of the event to be modified. Choose one of the following:
 - **Create Lead.**
 - **Update Contact**
 - **Update Lead**
6. Click **Edit** next to the external call you want to edit.



7. Review and edit the configuration details if necessary.
 - **Action:** The action that is performed and can include Create or Update.
 - **Entity:** The Salesforce entity that is affected in this call.

- **Trigger Salesforce Default Assignment Rule:** Enable only if you are using lead assignment rules in Salesforce. Typically, leads are assigned only upon lead creation.
 - **Send Email Notification:** Enable only if you have specific lead owners who need to receive Salesforce-generated email notifications when a new lead is assigned.
 - **External Call Return Value:** These options allow you to select which Oracle Eloqua field is used to store the reference ID returned by Salesforce. This needs to be specified only for the Create Lead external call.
8. Select the **Options** menu, then select **View Field Mapping**. Confirm that the fields are mapped as follows:

External Call	Salesforce Lead Fields	Oracle Eloqua Fields
Update Lead	Lead ID	SFDCLeadID
	All of the Salesforce fields for Create Lead	All of the Oracle Eloqua fields for Create Lead
Update Contact	Contact ID	SFDCContactID

9. If you need to create a field mapping, drag the corresponding Oracle Eloqua field from the *Eloqua Fields* column to the *Field Expressions* column.
10. Click **Save** after you complete the mappings for the external call.
11. If you are not implementing closed-loop reporting, also review the campaign association events:
 - **Associate Lead with Campaign**
 - **Associate Contact with Campaign**

These external calls should have the following field mappings:

External Call	Salesforce Lead Fields	Oracle Eloqua Fields
Associate Lead with Campaign	Campaign ID	Last SFDC Campaign ID
	Lead ID	SFDCLeadID
	Status	Last SFDC Campaign Status
Associate Contact with Campaign	Campaign ID	Last SFDC Campaign ID
	Contact ID	SFDCContactID
	Status	Last SFDC Campaign Status

Creating a custom campaign association external call

If you are not implementing closed-loop reporting, you must create an internal event that triggers both the *Create Lead* and *Associate Lead with Campaign* external calls. First the

Create Lead API call is executed, then after the Salesforce lead ID has been returned, the *Associate to Campaign* API call is executed.

Event Mapping

Overview - Use this page to configure the external calls that run against your CRM system as a result of your custom event being triggered in Eloqua.

Event Mapping

Custom Event Create Lead and Associate with Campaign

The above Eloqua event triggers the following external calls



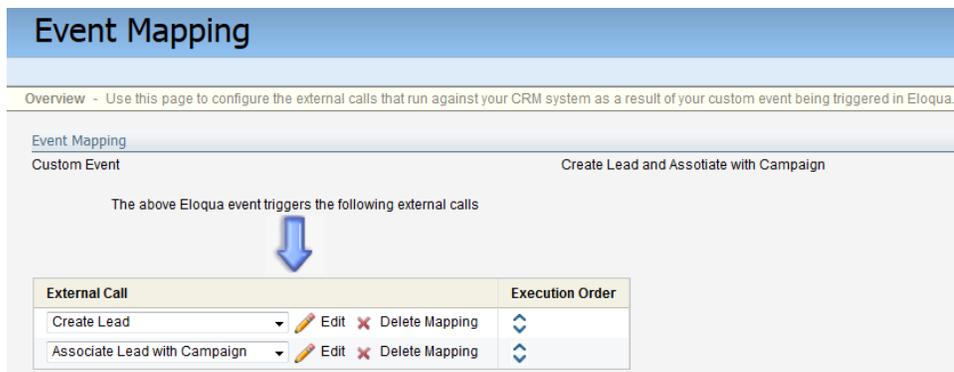
External Call	Execution Order
Create Lead ✎ Edit ✖ Delete Mapping	⬆
Associate Lead with Campaign ✎ Edit ✖ Delete Mapping	⬆

Note: You can only execute these Salesforce campaign association calls if you are not implementing closed-loop reporting. See [Closed-loop reporting with Salesforce](#) for more information on setting up these external calls for campaign associations.

To create a Salesforce campaign association call:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Outbound** tab.
4. In the *Internal Events* tab, expand the **Custom** folder under *All Internal Events*, then expand **Custom Contact Events**.
5. Click  next to *Custom Contact Events*, then click **Create New Custom Event**.

6. In the *Custom Event* window, enter the following **Custom Event Name**: Create Lead and Associate with Campaign.
7. Click **Save**. The new event appears in the navigation pane under *Custom Contact Events*.
8. Click the name of the new event.
9. From the *Event Mapping* window, click **Add Existing External Call** twice (you need to add two existing calls).
 - Select **Create Lead** as the first external call.
 - Select **Associate Lead with Campaign** as the second external call.



10. Click **Save**. A confirmation message appears if the save is successful.

Testing the external calls to Salesforce

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

After [configuring the external calls to Salesforce](#), it is recommended that you test the following external calls:

- Create Lead
- Update Lead
- Update Contact

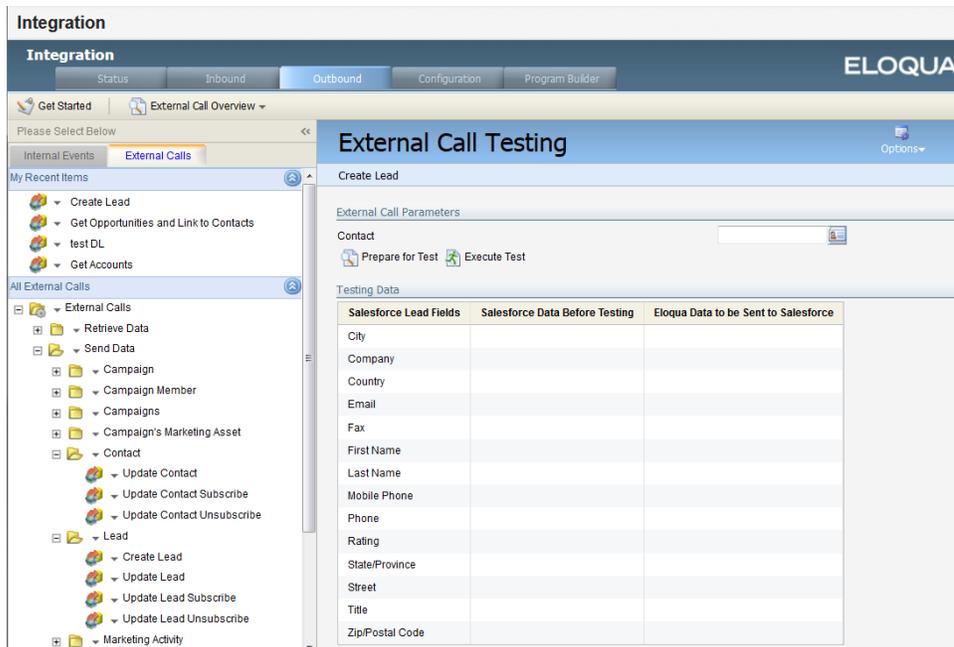
Use the External Call Testing tool to perform a field-by-field audit of how fields are populated and updated in Salesforce.

Important: Create a new Oracle Eloqua contact record to use for testing the Create Lead call. After successfully creating a new lead, make a change to the record in Oracle Eloqua before testing the Update Lead call so that you can confirm the update to the data.

To test outbound external calls:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Outbound** tab.
4. In the left panel, expand the **Send Data** folder under the *External Calls* tab, then expand both **Contact** and **Lead**.
5. Click  next to the *Create Lead* external call, then click **Test External Call**.

- Click  in the *External Call Testing* window to locate the Oracle Eloqua record you want to use for testing.



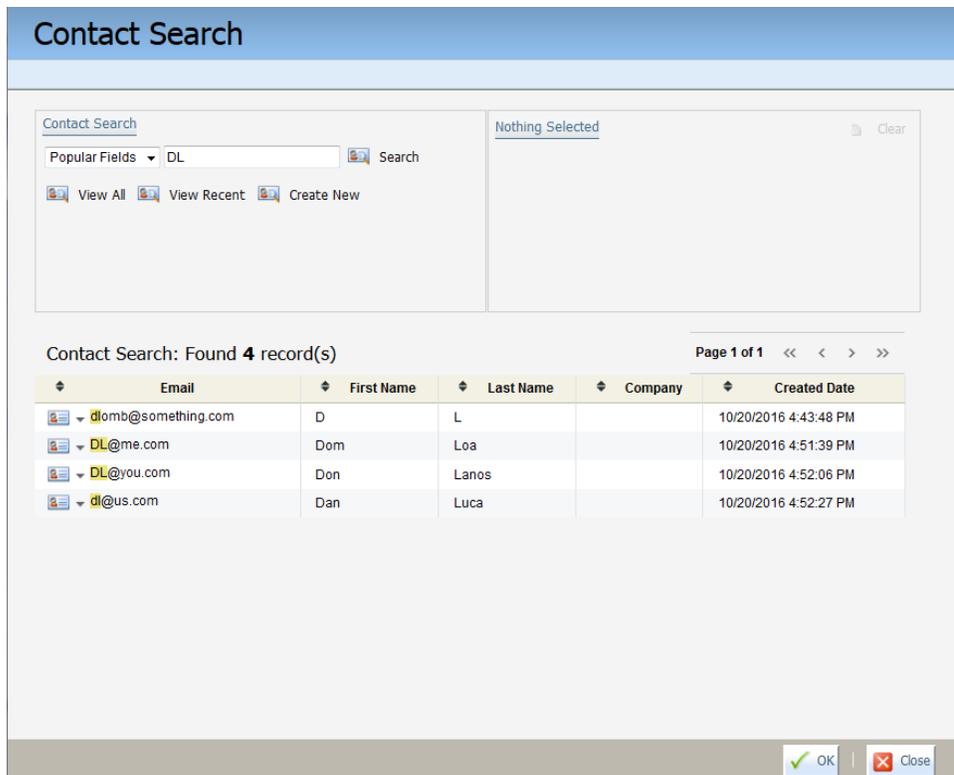
The screenshot shows the Oracle Eloqua Integration interface. The main window is titled "External Call Testing" and is part of the "Integration" section. The interface includes a navigation bar with tabs for "Status", "Inbound", "Outbound", "Configuration", and "Program Builder". The "Outbound" tab is active. Below the navigation bar, there are buttons for "Get Started" and "External Call Overview". The main content area is divided into two sections: "External Call Parameters" and "Testing Data".

The "External Call Parameters" section includes a "Create Lead" button and a "Contact" field. Below this, there are buttons for "Prepare for Test" and "Execute Test".

The "Testing Data" section contains a table with the following columns: "Salesforce Lead Fields", "Salesforce Data Before Testing", and "Eloqua Data to be Sent to Salesforce". The table lists various fields:

Salesforce Lead Fields	Salesforce Data Before Testing	Eloqua Data to be Sent to Salesforce
City		
Company		
Country		
Email		
Fax		
First Name		
Last Name		
Mobile Phone		
Phone		
Rating		
State/Province		
Street		
Title		
Zip/Postal Code		

- In the *Contact Search* window, enter the contact name and perform the search.
- Locate the record in the search results. Click  next to the contact's email address, then click **Select item**.



9. Click **OK**.
10. In the *External Call Testing* window, the selected contact's data is displayed in the table. Click **Prepare for Test** to view the state of the Salesforce database before and after the selected contact's data was sent.
11. Click **Execute Test** to complete the test.
12. Verify the results. The values in the *Eloqua Data to be Sent to Salesforce* column should match those in *Salesforce Data After Testing*. A banner message is displayed to indicate whether the test was successful.

Configuring Salesforce integration programs in Oracle Eloqua

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Important: This synchronization requires the use of programs built in Oracle Eloqua with Program Builder. If you're not familiar with programs in Oracle Eloqua, see [Program Builder](#) for an overview before continuing.

Oracle Eloqua programs automate when to trigger external calls to Salesforce. To integrate with Salesforce, Oracle Eloqua provides program templates that you will customize for your integration needs.

The following programs need to be setup to integrate Oracle Eloqua and Salesforce:

- **SYSTEM - CRM Email Opt Out program:** This program globally unsubscribes any Salesforce lead or contact that has the *Email Opt Out* flag checked in Salesforce from Oracle Eloqua. This ensures that your organization meets email compliance requirements and that the communication preferences of a record are synchronized across both systems. This program has standard logic and does not require additional configuration.

For information on enabling the *Email Opt Out* program, see [Enabling the email opt out program](#).

- **SYSTEM - CRM Update programs:** This is the primary integration program and determines how to update leads and contacts in Salesforce. There are two programs you can choose from: SYSTEM - CRM Update (Create only unique leads) and SYSTEM - CRM Update (Point of Interest).

For information on choosing and configuring the CRM Update program, see [Configuring the CRM update program](#)

Enabling the email opt out program

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

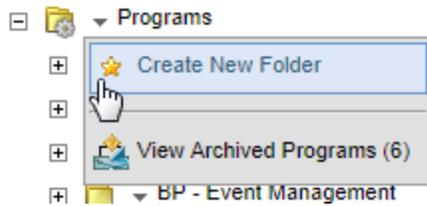
The SYSTEM - CRM Email Opt Out program globally unsubscribes any Salesforce lead or contact that has the *Email Opt Out* flag checked in Salesforce from Oracle Eloqua. This ensures that your organization meets email compliance requirements and that the communication preferences are synchronized across both systems. This program has standard logic and does not require configuration.

For an overview of the programs used in a Salesforce integration, see [Configuring Salesforce integration programs in Oracle Eloqua](#).

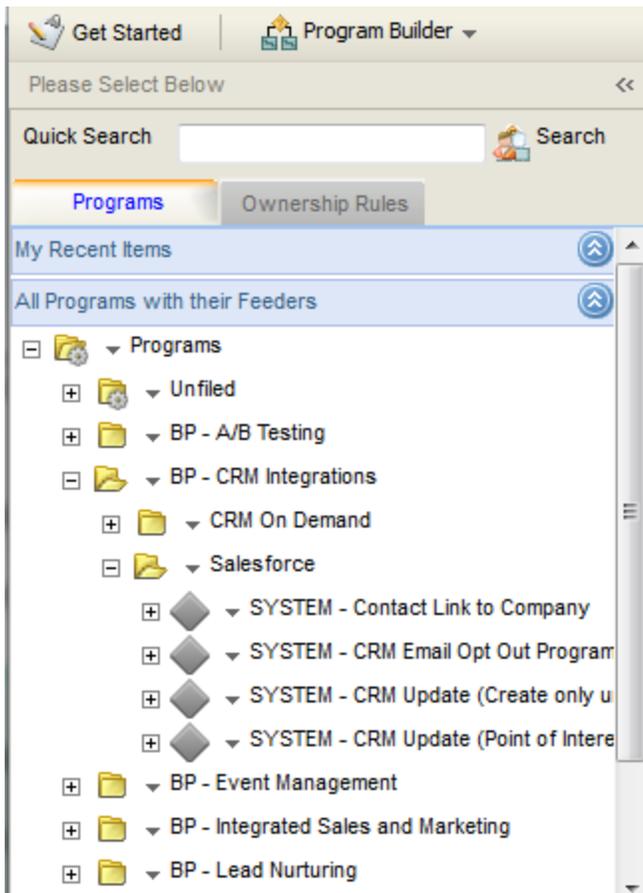
Note: The folders and paths discussed in this document are the defaults or recommendations. Your environment may use different names.

To enable the SYSTEM - CRM Email Opt Out program:

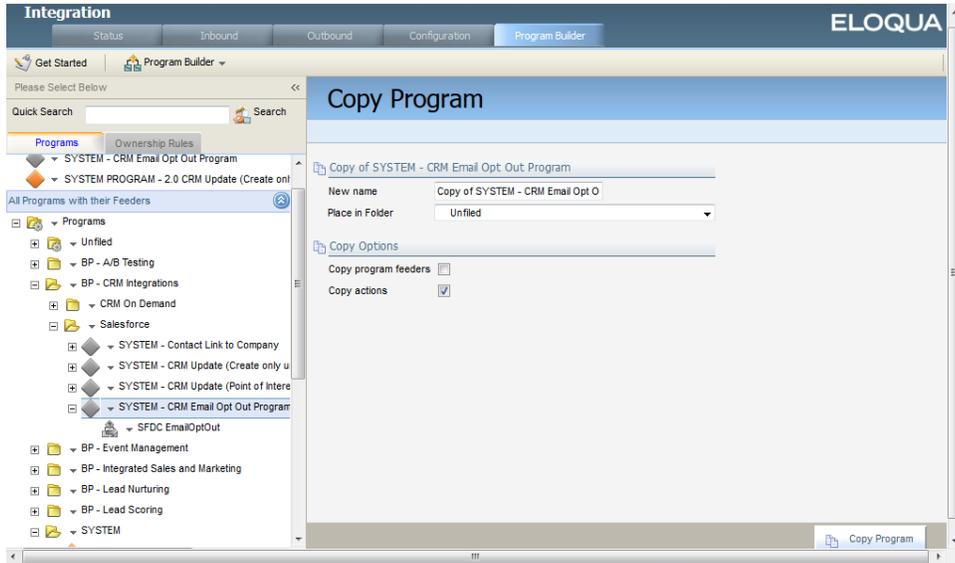
1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Program Builder** tab.
4. In the *Programs* tab, if a SYSTEM folder does not already exist, create the folder. To create a new folder, in the folder list, click  next to *Programs*, then click **Create New Folder**.



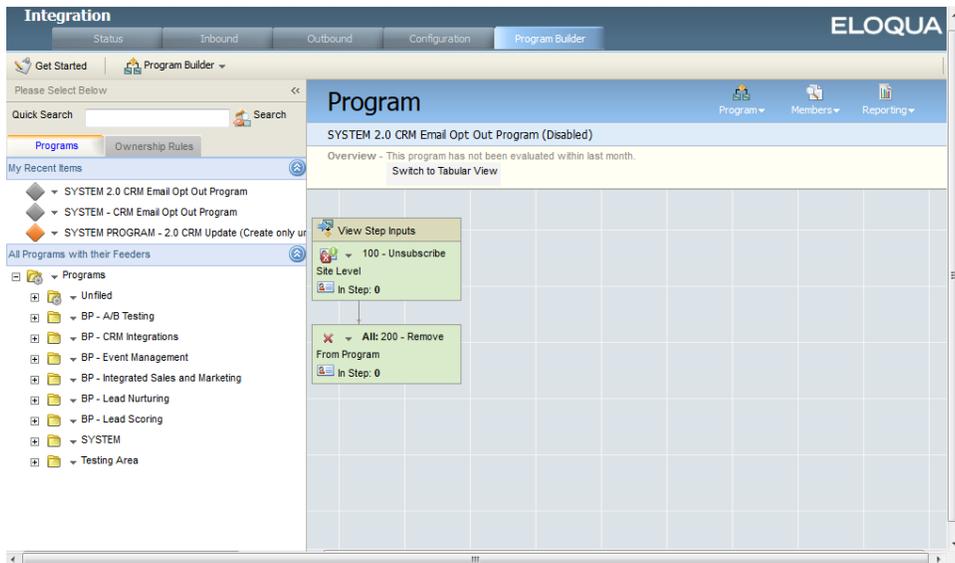
5. Navigate to the **BP - CRM Integrations** folder then open the **Salesforce** folder.



6. Copy **SYSTEM - CRM Email Opt Out Program** to the **SYSTEM** folder.
 - a. Click **SYSTEM - CRM Email Opt Out Program**.
 - b. Click the **Program** menu, then click **Copy Program**.



- c. Enter the name **System 2.0 SFDC Email Opt Out** and select the **SYSTEM** folder from the *Place in Folder* drop-down.
- d. Enable both of the options **Copy program feeders** and **Copy actions**.
- e. Click **Copy Program**.



7. Navigate to program that you created. It should now appear in the *My Recent Items* list.
8. Click the **Program** menu, then click **Enable Program**.
9. If you are prompted to, enable the appropriate run mode based on the descriptions provided onscreen.

 To enable this program, please select the run mode below that best suits your needs.

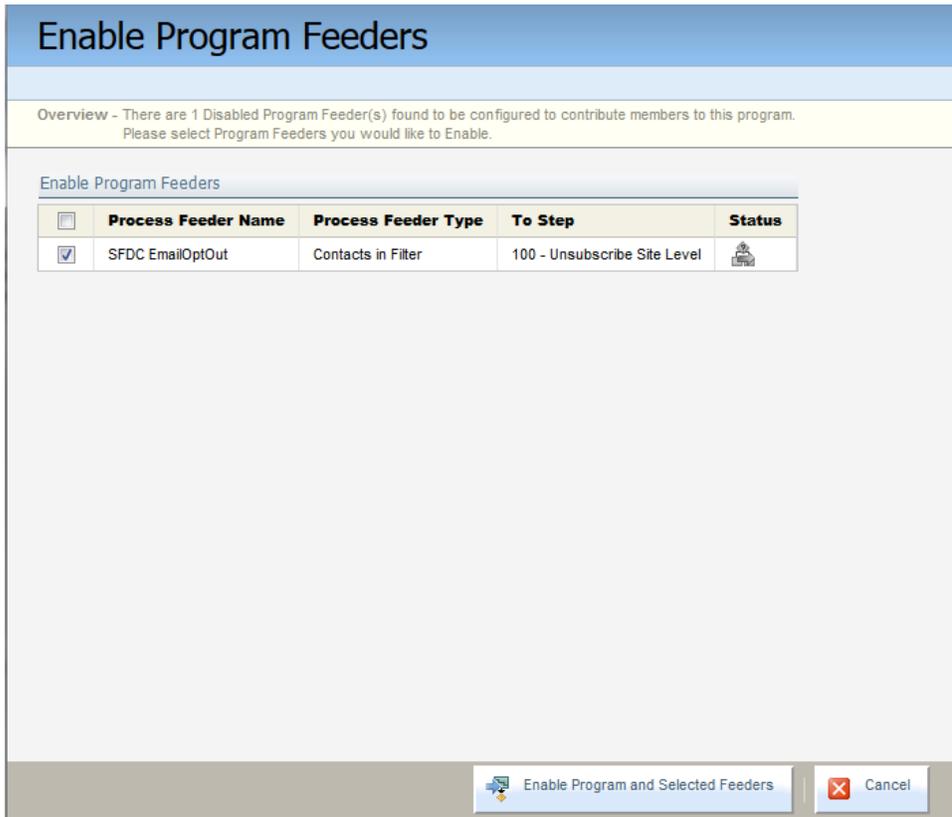
Standard Mode	Priority Mode	Bulk Mode
For programs that use all types of marketing actions and decisions. This mode works best for general purpose automation with no special requirements.	For programs that are updating data and have no loops or complex logic. This mode is best for programs that need to execute quickly (after a form submission for example)	For drip marketing programs, batch updating of many records, and complex programs that have many steps
 15 Minute Evaluation Time	 5 Minute Evaluation Time	 2 Hour Evaluation Time
 100,000 Contacts per Hour	 3,000 Contacts per Hour	 50,000 Contacts per Hour
<input type="button" value="Enable"/>	<input type="button" value="Enable"/> <small>5 program(s) are already in priority mode. The maximum allowed is 5</small>	<input type="button" value="Enable"/>

Test Mode

No information is available about who last changed the run mode of this program.

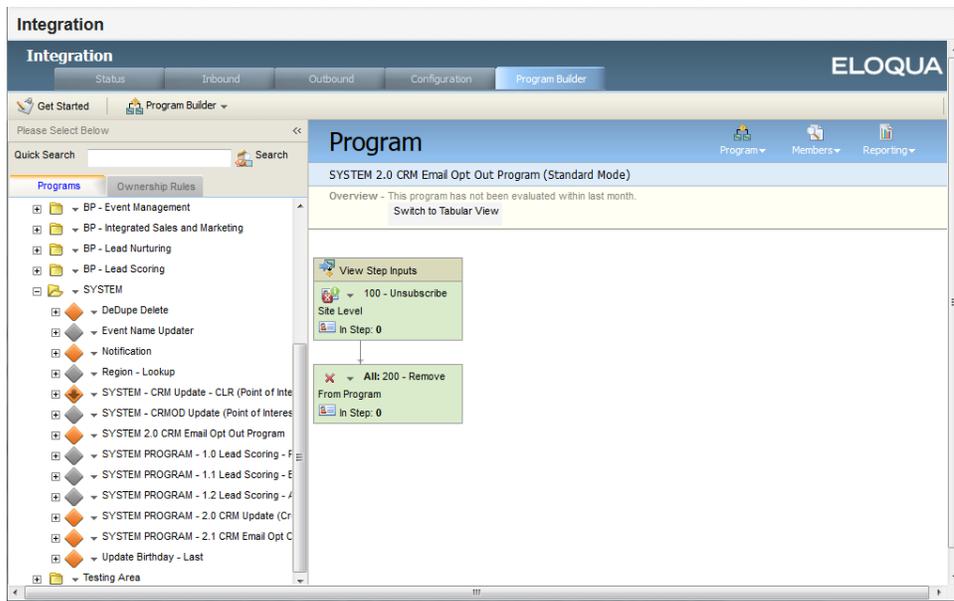
 **Note:** It's recommended that you use standard mode for this.

10. In the *Enable Program Feeders* window, select the check box next to the feeder.



11. Click **Enable Program and Selected Feeders**.

Note: Verify that the program is active by confirming that the program workflow background is white and the program icon is no longer gray.



Configuring the CRM update program

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the [Salesforce Integration app](#) in its place. Learn more in [our product notice](#).

The SYSTEM - CRM Update program is the primary integration program and determines how to update leads and contacts in Salesforce. There are two program templates you can choose from:

- **SYSTEM - CRM Update (Create only unique leads) program:** Creates a new lead record in Salesforce only if there is no existing lead or contact with the same email address. If there is an existing lead or contact, the program will update the existing entity instead of creating a new lead.

- **SYSTEM - CRM Update (Point of Interest) program:** Creates a new lead record even if there is an existing lead or contact in Salesforce CRM.

Choose one of these programs to use for your Salesforce integration then configure the program according to the details provided in below.

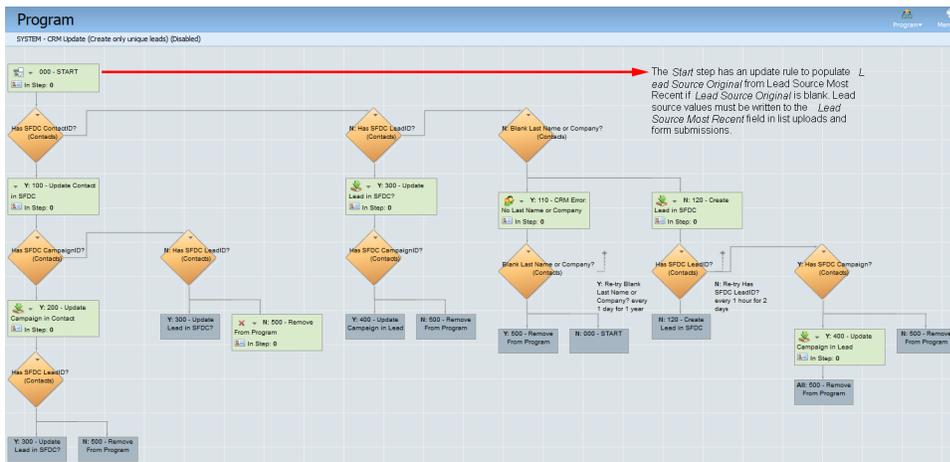
Note: If you do not have these program templates in your Oracle Eloqua instance, contact [My Oracle Support](https://support.oracle.com) (<https://support.oracle.com>).

Choosing the right CRM update program

Before you configure your CRM Update program you must choose a program template to use for your CRM update program.

About the SYSTEM - CRM Update (Create only unique leads) program

This program creates a new lead record in Salesforce only if there is no existing lead or contact with the same email address. If there is an existing lead or contact, the program will update the existing entity instead of creating a new lead.

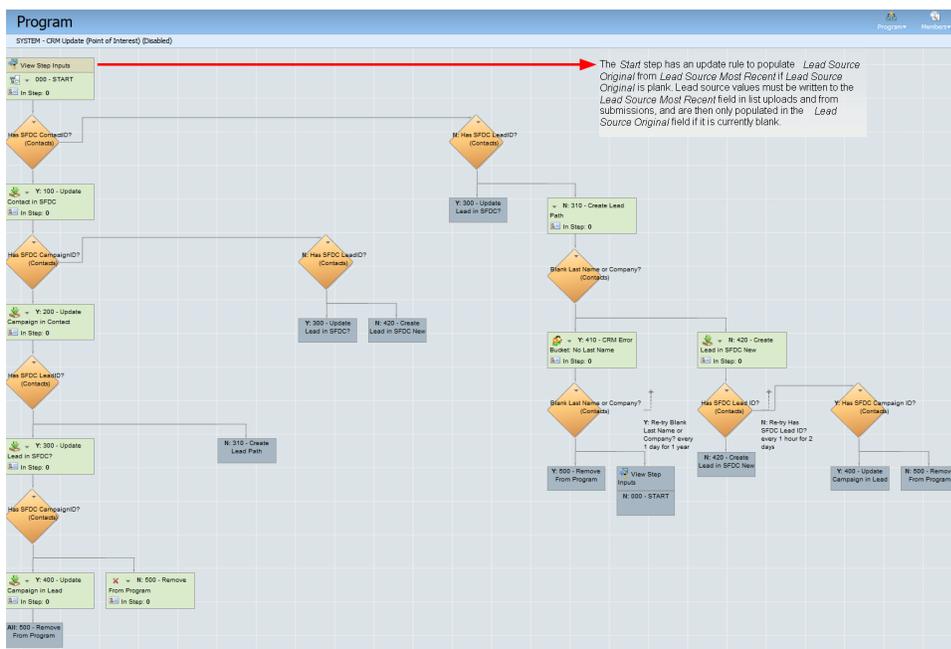


This program uses the following logic:

Has lead ID on Eloqua contact record?	Has contact ID on Eloqua contact record?	Action
No	No	Create Lead in Salesforce
Yes	No	Update Lead in Salesforce
Yes	Yes	Update Lead and Contact in Salesforce
No	Yes	Update Contact in Salesforce

About the SYSTEM - CRM Update (Point of Interest) program

This program creates a new lead record (or updates existing lead if one exists) even if there is an existing contact in Salesforce.



This program uses the following logic:

Has lead ID on Eloqua contact record?	Has contact ID on Eloqua contact record?	Action
No	No	Create Lead in Salesforce
Yes	No	Update Lead in Salesforce
Yes	Yes	Update Lead and Contact in Salesforce
No	Yes	Create Lead and Update Contact in Salesforce

Configuring the selected SYSTEM - CRM Update program

It's recommended that you map no more than 50 fields in the update process to from Oracle Eloqua to Salesforce.

Note: The folders and paths discussed in this document are the defaults or recommendations. Your environment may use different names.

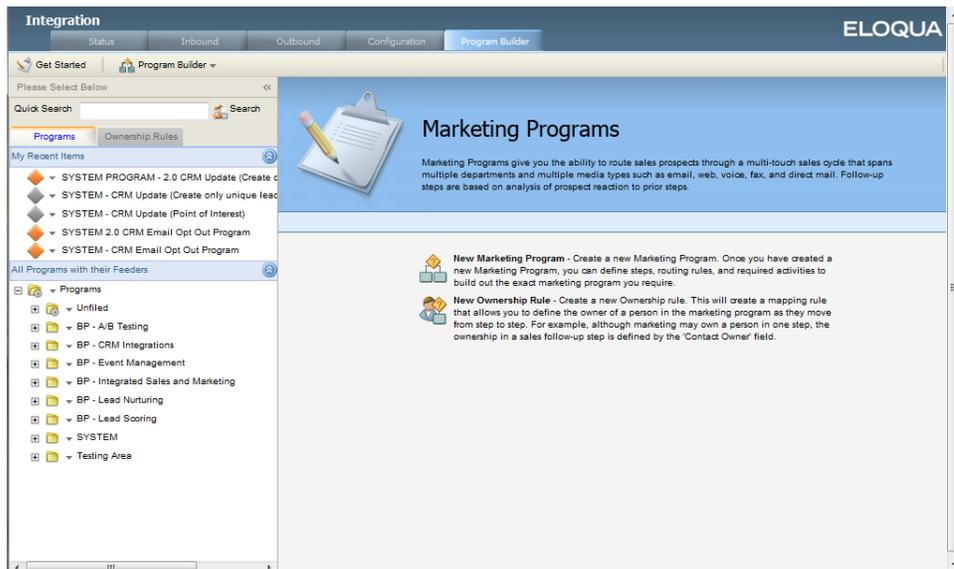
Before you begin:

- Choose the appropriate program template to use for your integration. Refer to [About the SYSTEM - CRM Update \(Create only unique leads\) program](#) and [About the SYSTEM - CRM Update \(Point of Interest\) program](#) for more information.
- [Configuring external calls to send data to Salesforce.](#)

Configuring the SYSTEM - CRM Update program:

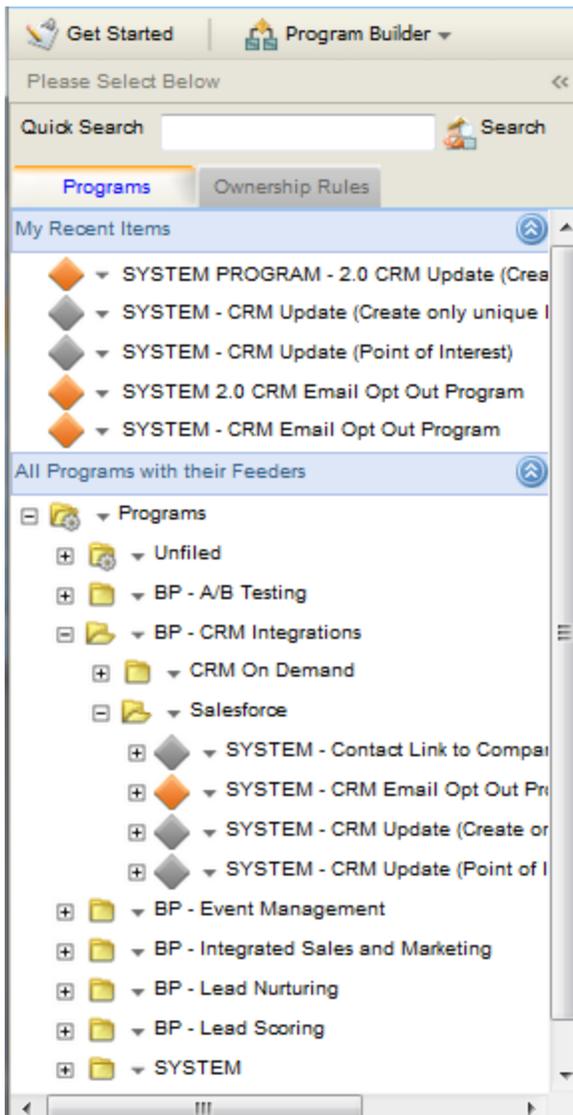
1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.

3. Click the **Program Builder** tab.

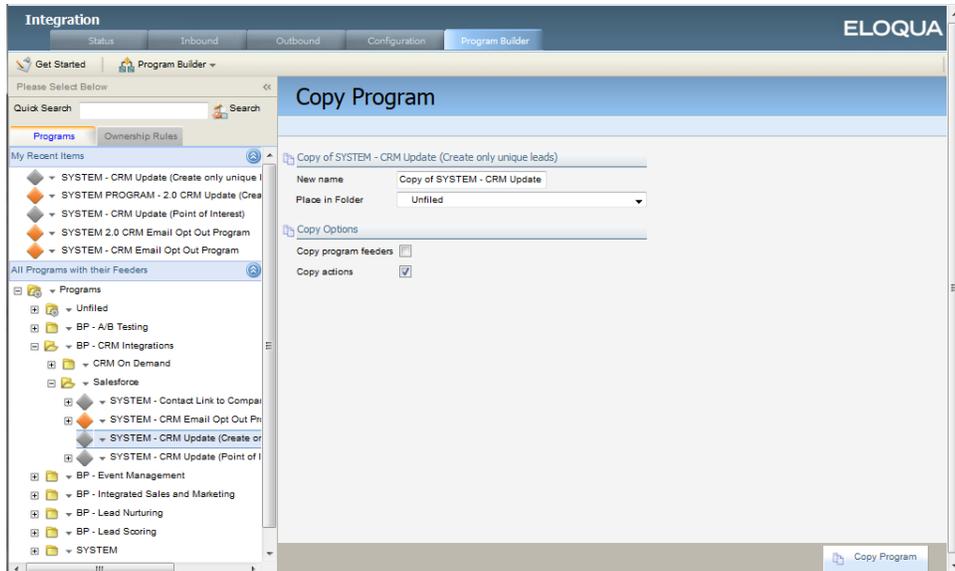


4. In the *Programs* tab, if a **SYSTEM** folder does not already exist, create the folder. To create a new folder, in the folder list, click  next to *Programs*, then click **Create New Folder**.

5. Navigate to the **BP - CRM Integrations** folder then open the **Salesforce** folder.

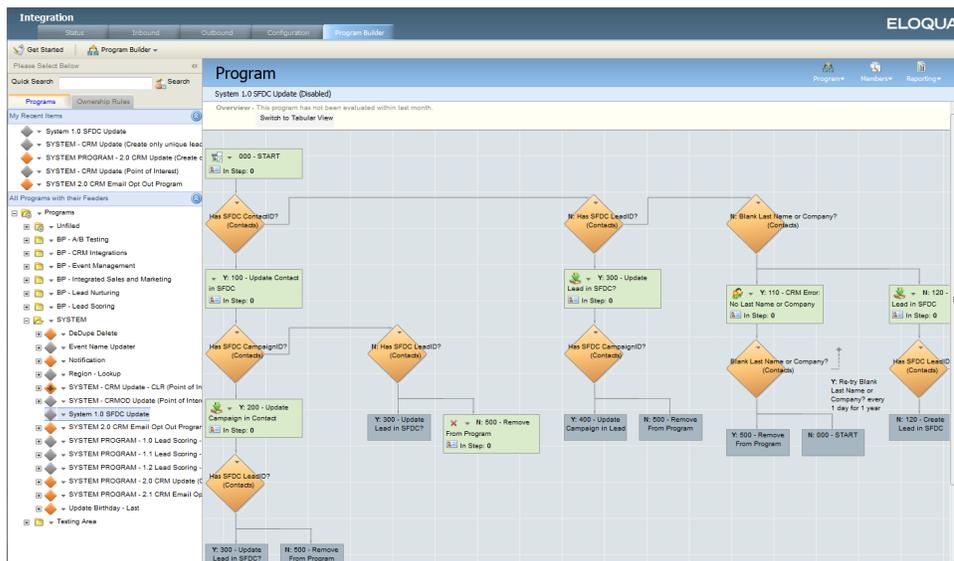


6. Locate the CRM Update program you have selected.
 - SYSTEM - CRM Update (Create only unique leads)
 - SYSTEM - CRM Update (Point of Interest)
7. Copy the CRM update program to the SYSTEM folder.
 - a. Click the program name.
 - b. Click the **Program** menu, then click **Copy Program**.



- c. Enter the name **System 1.0 SFDC Update** and select the **SYSTEM** folder from the *Place in Folder* drop-down.
- d. Enable both of the options **Copy program feeders** and **Copy actions**.
- e. Click **Copy Program**.

8. Navigate to program that you created. It should now appear in the *My Recent Items* list.



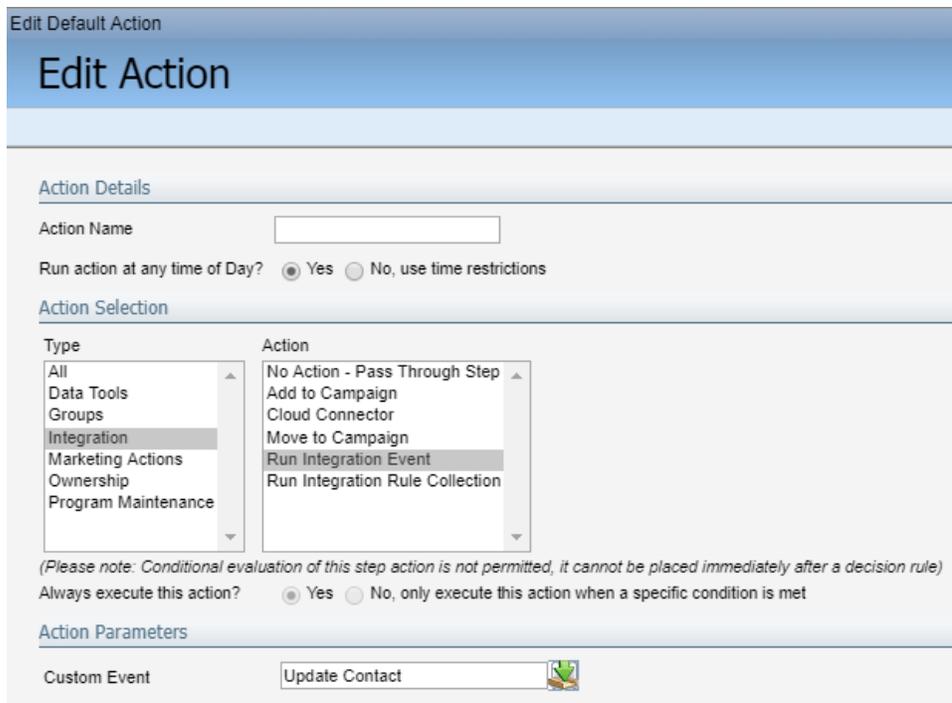
9. For each of the following steps, click  next to the step name, then click **Edit Step Default**

Action to configure the settings:

- Update Contact in SFDC
- Update Lead in SFDC
- Create Lead in SFDC

10. Configure the settings for each step as follows:

- The action can run at any time of day.
- Select **Integration** as the action type and **Run Integration Event** as the action.
- Select the **Custom Event** to run. The custom event you choose depends on the step you are updating. For example, the *Update Contact in SFDC* step should run the *Update Contact* event. For more information on these events, see [Configuring external calls to send data to Salesforce](#).



Edit Default Action

Edit Action

Action Details

Action Name

Run action at any time of Day? Yes No, use time restrictions

Action Selection

Type	Action
All	No Action - Pass Through Step
Data Tools	Add to Campaign
Groups	Cloud Connector
Integration	Move to Campaign
Marketing Actions	Run Integration Event
Ownership	Run Integration Rule Collection
Program Maintenance	

(Please note: Conditional evaluation of this step action is not permitted, it cannot be placed immediately after a decision rule)

Always execute this action? Yes No, only execute this action when a specific condition is met

Action Parameters

Custom Event 

11. Click **Save and Close**.

12. Choose an option:

- If you are not implementing closed-loop reporting, update the campaign association steps to execute the applicable campaign association events. The steps you have to update will vary depending on the

program template you are using. For example, the, *Update Campaign in Lead* and *Update Campaign in Contact* should execute the Campaign Custom Contact Internal Events *Associate Lead With Campaign* and *Associate Contact With Campaign* events. For more information on these events, see [Configuring external calls to send data to Salesforce](#).

- If you are implementing closed-loop reporting, See [Closed-loop reporting with Salesforce](#) for more information on setting up the program.

13. After configuring all the steps, click the **Program** menu, then click **Enable Program**.
14. If you are prompted to, enable the appropriate run mode based on the descriptions provided onscreen.

To enable this program, please select the run mode below that best suits your needs.

Run Mode	Description	Evaluation Time	Contacts per Hour
Standard Mode	For programs that use all types of marketing actions and decisions. This mode works best for general purpose automation with no special requirements.	15 Minute	100,000
Priority Mode	For programs that are updating data and have no loops or complex logic. This mode is best for programs that need to execute quickly (after a form submission for example)	5 Minute	3,000
Bulk Mode	For drip marketing programs, batch updating of many records, and complex programs that have many steps	2 Hour	50,000
Test Mode			

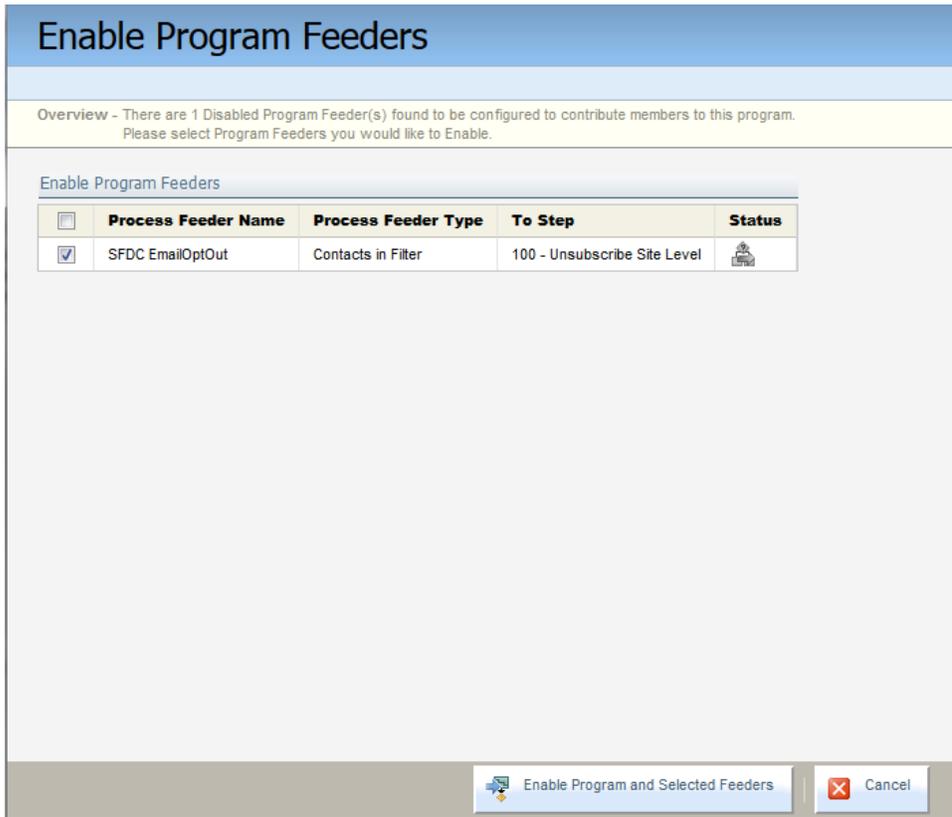
5 program(s) are already in priority mode. The maximum allowed is 5

No information is available about who last changed the run mode of this program.

Close

Note: It's recommended that you use priority mode for this program.

15. In the *Enable Program Feeders* window, check the box next to the feeder.



16. Click **Enable Program and Selected Feeders**.

Note: Verify that the program is active by confirming that the program workflow background is white and the program icon is no longer gray.

Enabling Salesforce activity writing

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Activity writing allows Oracle Eloqua-tracked activities, such as email clickthroughs, website visits, and form submissions, to be written to Salesforce as closed tasks. These tasks are associated with the Salesforce contact or lead who performed the activity. Because these tasks can be resource-intensive in terms of Salesforce storage space, it is important to determine which ones provide the most useful information.

We recommend handling the following activities:

- Email bounceback
- Email clickthrough
- Email open
- Email subscribe
- Email unsubscribe
- Form submit
- Website visit

These activities are written to Salesforce in a synchronous queue and require no configuration other than enabling them. Should there be a temporary problem with the destination Salesforce system, Oracle Eloqua stores activities in a queue, which is processed when Salesforce access issues have been resolved.

To enable Salesforce activity writing in Oracle Eloqua:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Outbound** tab.
4. Expand the **Activity** folder. All available internal events (activities) are listed.
5. For each of the Oracle Eloqua activities that you want to be sent to Salesforce, click  next to the internal event name and enable it from the drop-down menu.

These are the events that you are recommended to enable:

- Email bounceback
- Email clickthrough
- Email open
- Email subscribe
- Email unsubscribe
- Form submit
- Website visit

Important: Do not enable *Email send*. This activity generates significant volume and does not provide useful information. If you want to share information about email sends, you can use an Oracle Eloqua report or a Profiler report (Profiler provides you information in real-time).

The status of the activity is indicated by the icon displayed before its name:

-  (green) - The activity is currently enabled.
-  (gray) - The activity is disabled.

Tip: After you enable the required activities, it is good practice to test them to ensure the activity is being written to Salesforce as closed tasks.

Initializing and monitoring the Salesforce integration

Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

Auto synchs and queues were disabled during the [preliminary setup process](#) for this integration. With the required synchs and queues created and the integration configurations completed, you can now re-initialize the system and configure notifications for the ongoing monitoring and maintenance of the integration.

This section includes the following tasks:

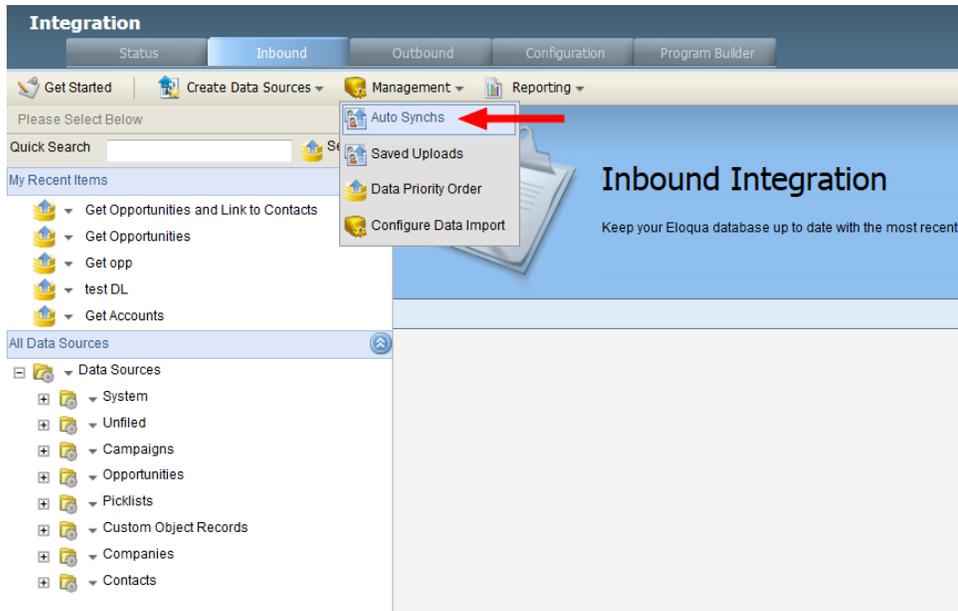
- [Re-enabling the auto synchs](#)
- [Enabling the internal and external queues](#)
- [Configuring system error notifications](#)

Re-enabling the auto synchs

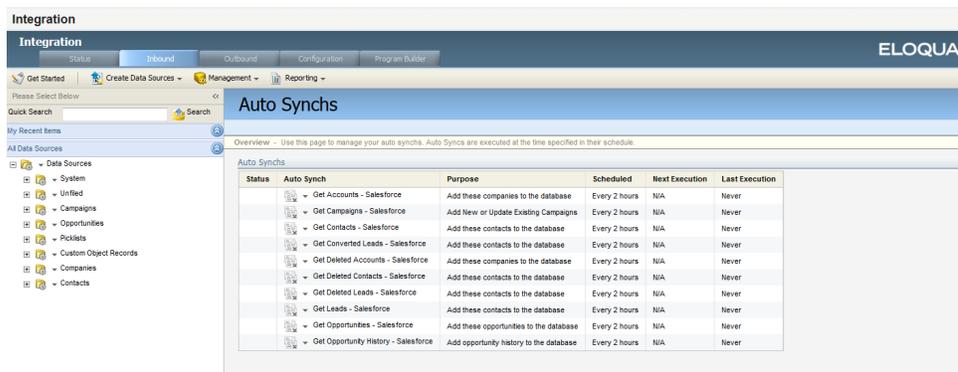
Now that you have completed the tasks for [configuring data synchronization](#) from Oracle Eloqua to Salesforce, you can re-enable the auto-synchs previously disabled.

To enable auto synchs:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Inbound** tab.
4. Select **Auto Synchs** from the *Management* menu.



5. Click the **Get Deleted Accounts** auto synch in the *Auto Synch* column. All the existing auto synchs are listed. A green check mark beside an auto synch indicates that it completed successfully the last time it ran.



6. In the *Auto Synch* window, click **Edit** for the *Auto Synch Details* group.

7. Check the **Enabled** box in the *Upload Details* window.
8. For notification purposes, update the *Email* field with the address of your Salesforce or Oracle Eloqua administrator. You can specify multiple recipients using a semicolon (;) as a separator. You can update the address at any time.
9. You can enable the following notification options:
 - **Send Email on Success:** This sends a notification email each time the auto synch is executed successfully. Although these notifications are not mandatory on an ongoing basis, it is a good idea to select this initially as you complete and test the Salesforce integration.
 - **Send Email on Failure:** This sends a notification email each time the auto synch fails. The notification explains the cause of failure (for example, inaccessible Salesforce fields, or changes in CRM user password). This notification should always be enabled.
10. Click **Save and Close**.
11. Click **Close** in the *Auto Synch* window.
12. If you want to run the auto synchs now, rather than wait for the scheduled upload, click  next to the auto synch name, then click **Run Auto Synch**.
13. Repeat this process for each of the seven integration auto synchs.

Important: This is the recommended order for enabling the remaining auto synchs: Get Accounts, Get Deleted Leads, Get Converted Leads, Get Leads, Get Deleted Contacts, Get Contacts.

Enabling the internal and external queues

As part of the initial preparations for this integration, you **disabled both internal events and external calls**. You can now re-enable the queues or confirm that they are already enabled.

To enable the internal and external queues:

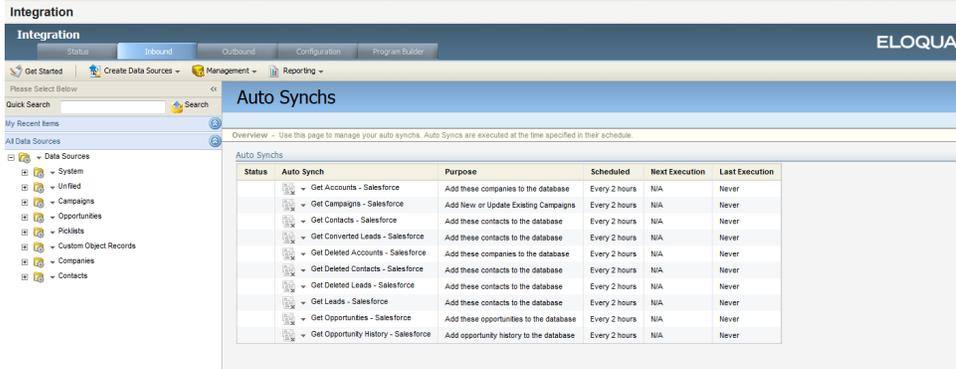
1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. In the *Status* tab, click the **Reporting** menu, then click **Integration Reporting**.
4. Click **Enable** beside both the *Internal Event Queue Stats* and *External Call Queue Stats* headings. The option to enable appears only if you previously disabled the queue. If *Pause* is currently displayed, the queue is already enabled

Validating auto synch processing

After you have configured the account, contact, and lead auto synchs, verify that they are running correctly.

To test your auto synchs:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Inbound** tab.
4. In the *Management* menu, click **Auto Synchs**. All the existing auto synchs are listed. A green check mark beside an auto synch indicates that it was successfully completed the last time it ran.



5. To view historical processing data for an auto synch, click on the name of the auto synch in the *Auto Synchs* area.

6. From the *Reporting* menu, click **Auto Synch History**.

A report is displayed that lists status details for each run of the auto synch. You can:

- Print the report or export it to Excel by clicking the options in the *Export* menu.
- View more detailed information about a specific auto synch processing run by selecting the down arrow in the first column of the table, then clicking **View Upload Details**.

Configuring system error notifications

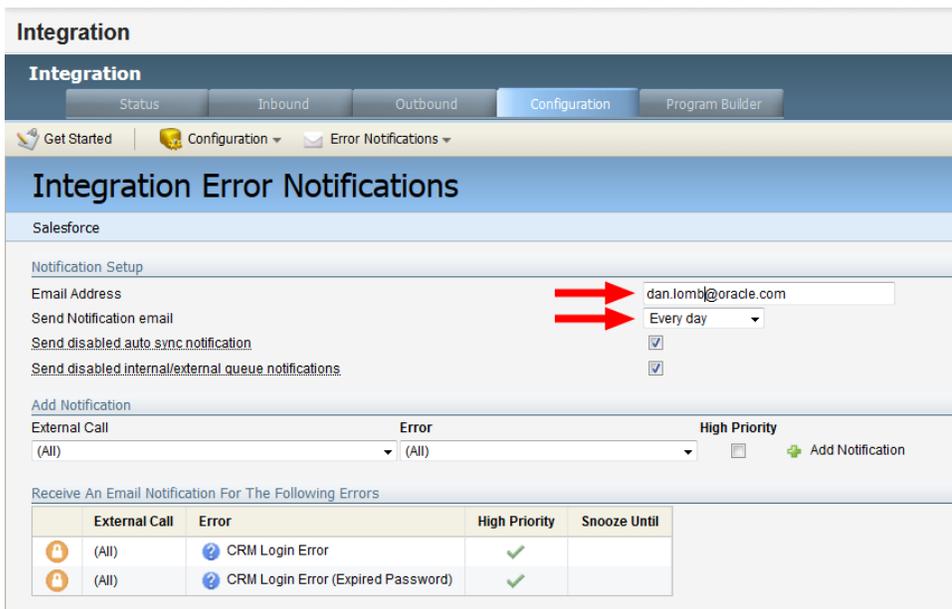
With both the integration auto synchs and the internal and external queues re-enabled, you can now set up error notifications.

Errors can cause issues with the Salesforce integration. Configure error notifications to send alert emails when specified errors occur. Determine who on your team should receive these notifications on an on-going basis. These are the recommended errors to trigger notification:

External Call: (All)	Error: CRM Login Error
External Call: Create Lead	Error: (All)
External Call: Update Lead	Error: (All)
External Call: Update Contact	Error: (All)

To configure error notifications:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. In the *Configuration* tab, select **Manage Error Notifications** on the *Error Notifications* menu.
The Integration Error Notifications window opens.
4. In the *Notification Setup* area, enter the **Email Address** for the notification recipient and select the frequency with which the email will be sent.



Integration

Integration

Status Inbound Outbound Configuration Program Builder

Get Started Configuration Error Notifications

Integration Error Notifications

Salesforce

Notification Setup

Email Address

Send Notification email

Send disabled auto sync notification

Send disabled internal/external queue notifications

Frequency: Every day

Add Notification

External Call	Error	High Priority	Snooze Until
(All)	CRM Login Error	<input checked="" type="checkbox"/>	
(All)	CRM Login Error (Expired Password)	<input checked="" type="checkbox"/>	

5. In the *Add Notification* area, select the external call and the specific error that should trigger a notification. It is typically a good idea to select **(All)** for the error type. You can also select whether this is a high priority error, which overrides the defined schedule and sends a notification every hour.
6. Click **Add Notification**. The notification appears in the *Receive An Email For The Following Errors* table.

7. Repeat to add more error notifications.
8. Click **Save**.

Resetting the Salesforce password in Oracle Eloqua

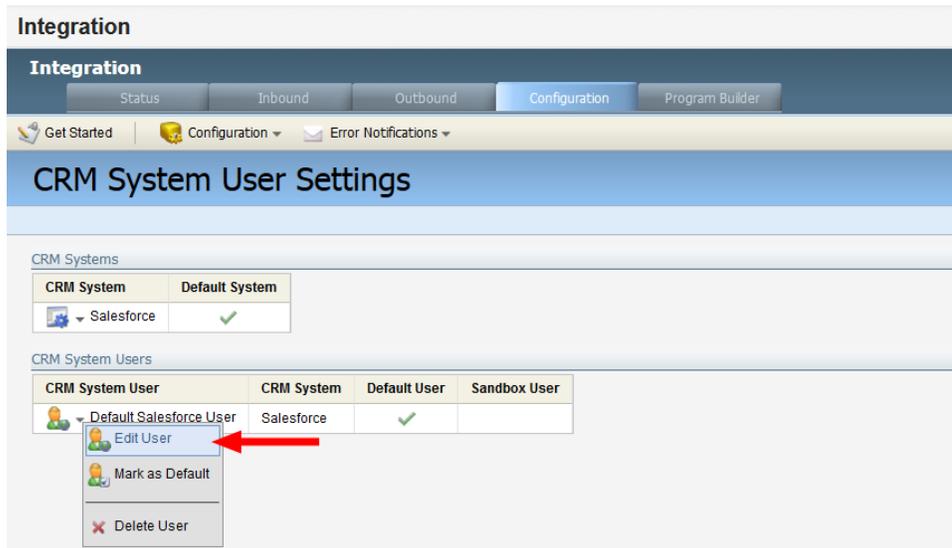
Important: The Salesforce native integration was deprecated February 1, 2021. We recommend using the the [Salesforce Integration app in its place](#). Learn more in [our product notice](#).

If the user set up for the ongoing data exchange between Oracle Eloqua and Salesforce was set up with a password expiration, then you will need to reset the password when it expires. For more information on this user account, see [Preliminary Salesforce setup for your integration](#).

To update an expired Salesforce password in Oracle Eloqua:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Configuration** tab.
4. Click **Manage CRM System Users** in the *Configuration* menu.

- In the *CRM System Users* area, click  next to *Default Salesforce User*, then click **Edit User**.



- Click **Change Password** in the *CRM System User* window.
- In the *Change Password* window, enter the new password in the **New Password** and **Re-Type Password** boxes.
- Click **Change**.
- Click **Save and Close**.