



# Oracle Eloqua and Salesforce Closed-Loop Reporting

## Configuration Guide

# Contents

---

- Closed-loop reporting with Salesforce** ..... 4
- Steps to configure closed-loop reporting** ..... 7
- Preliminary setup for closed-loop reporting** ..... 9
- Creating campaign response rules for closed-loop reporting** ..... 11
- Configuring campaign revenue settings for closed-loop reporting** ..... 15
  - Creating the Salesforce opportunity stages ..... 15
  - Configuring the default revenue attribution method ..... 18
- Configuring auto synchs for closed-loop reporting** ..... 20
  - Creating the Get Campaigns auto synch ..... 21
  - Creating the Get Opportunities auto synch ..... 27
  - Creating the Get Opportunities auto synch to associate opportunities with contacts ..... 31
  - Disabling auto synchs for closed-loop reporting ..... 38
- Configuring external calls for closed-loop reporting** ..... 40
  - Preconfiguration: Creating folders ..... 41
  - Creating external calls ..... 42
- Sending campaign data to Salesforce for closed-loop reporting** ..... 48
  - Filtering contacts that responded to campaigns ..... 49
  - Creating the send campaign information program ..... 50
- Updating the CRM update program for closed-loop reporting** ..... 58
- Initializing closed-loop reporting** ..... 60
  - Enabling the auto synchs ..... 60
  - Enabling the internal events ..... 61
  - Validating auto synch processing ..... 61

---

Configuring system notifications about errors .....	62
Closed-loop reports .....	64

# Closed-loop reporting with Salesforce



**Important:** Typically organizations work with Oracle or with Oracle partners to ensure a successful CRM integration. Find out more about Oracle's [implementation services](#).

This documentation provides you with 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.

Closed-loop reporting allows you to attribute sales opportunity revenue to marketing campaigns so that you can analyze your overall campaign revenue performance (ROI), and identify the best and worst performing campaigns. In a Salesforce and Oracle Eloqua integration, closed-loop reporting allows you to monitor opportunities and sales against campaign responses. This allows you to attribute opportunities and closed-won revenue back to your campaign activities and costs, and gives you a more complete understanding of campaign performance and ROI.

[Learn more by watching the video](#)

To implement closed-loop reporting with a Salesforce and Oracle Eloqua integration, there are several configuration tasks that you must complete. After implementation, you can use closed-loop reports in Insight to demonstrate real ROI and revenue impact from your marketing campaigns. While you can set up closed-loop reporting in parallel with the Salesforce integration, consider implementing after the primary integration is complete. This allows you to ensure that the Salesforce integration works as expected before you introduce additional data synchronization.

**Note:** You do not have to implement closed-loop reporting functionality. Configuring closed-loop reporting is an optional task.

Consider the benefits of using Oracle Eloqua campaigns for closed-loop reporting:

- Create, execute, and maintain your campaigns in one system versus multiple systems.
- Save on resources, time, training, and expenses.
- Generate report views that include metrics for all of your marketing assets, such as forms, landing pages, and emails.
- Automatically send required campaign information to Salesforce so that your Sales and Customer Service teams can access it.
- Evaluate ROI reports to determine the effectiveness of your marketing campaigns in generating revenue.

[Learn more by watching the video](#)

# Campaign costs

Before implementing closed-loop reporting, develop a consistent way to determine campaign costs. Understanding the actual costs for a campaign is a very important step in enabling closed-loop reporting. Without the costs, the ROI reports are unable to calculate a return value against the campaign investment. Establish a method for calculating the cost of each campaign type. Consistently use that costing methodology across all campaign owners in the organization so that your measurements are consistent and you can get a better performance comparison.

The most important aspect when determining your costs is keeping your approach consistent. If you are just getting started with closed-loop reporting, consider simply including all discretionary costs spent on campaign development and execution. This can include sponsorship fees dedicated to demand generation efforts, infrastructure costs from discretionary budgets, and outsourced creative, imagery, or copy writing support.

Take a quarterly thought-leadership webinar as an example. You may include the follow costs:

- Webinar host provider
- Thought leader speaker fee
- Creative budget (imagery, copy writing for campaign assets)
- Online ad space dedicated to promotion

# Steps to configure closed-loop reporting

The following table provides an overview of the steps you need to configure closed-loop reporting with Salesforce:

Step	Description
Preliminary setup	Create the necessary campaign fields and external assets.  See <a href="#">Preliminary setup for closed-loop reporting</a>
Configure campaign response rules	Create response rules that defines campaign activities that are important to your organization and will be used to update the Salesforce campaign member status.  See <a href="#">Creating campaign response rules for closed-loop reporting</a>
Configure campaign revenue settings	Setup the Salesforce opportunity stages in Oracle Eloqua and configure how to attribute revenue to campaigns.  See <a href="#">Configuring campaign revenue settings for closed-loop reporting</a>
Configure the auto synchs	Configure the auto synchs used to import campaign and opportunity data from Salesforce.  See <a href="#">Configuring auto synchs for closed-loop reporting</a>
Configure external calls	Configure the external calls to export campaign data and campaign associations to Salesforce.  See <a href="#">Configuring external calls for closed-loop reporting.</a>

Step	Description
Configure the programs	<p>Configure the program used to send campaign data to Salesforce and update the main CRM update program.</p> <p>See <a href="#">Sending campaign data to Salesforce for closed-loop reporting</a> and <a href="#">Updating the CRM update program for closed-loop reporting</a>.</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 monitor and maintain the integration.</p> <p>See <a href="#">Initializing closed-loop reporting</a>.</p>



# Preliminary setup for closed-loop reporting

Before implementing closed-loop reporting with Salesforce, you must perform the following in Oracle Eloqua:

- [Creating campaign fields](#)
- [Creating external assets and activities](#)

## Creating campaign fields

Custom campaign fields allow you to mimic the campaign structure in your Salesforce system. When a campaign is created in Oracle Eloqua, the custom campaign fields are synchronized to Salesforce. Similarly, changes to the campaign fields in Salesforce are synchronized into Oracle Eloqua.

The following table lists the default campaign fields for Oracle Eloqua campaigns and their data types. An asterisk (\*) denotes required fields in Oracle Eloqua.

Default campaign fields Data type	
Campaign Name*	Text
Start Date*	Date
End Date*	Date
SFDC Campaign ID	Text
Budgeted Cost	Currency
Actual Cost	Currency
Region	Picklist
Product	Picklist
Campaign Type	Picklist

If you've identified the need for additional custom campaign fields, see [Creating custom campaign fields](#). The custom fields should be setup to allow users to configure them in a way expected by Salesforce. For example, if Salesforce is expecting a date, set the campaign field up to only allow for dates.

## Creating external assets and activities

External activities are those activities that are not tracked by Oracle Eloqua. For example, trade show attendance is not tracked by Oracle Eloqua, while an email open is. External activities help give your organization a full view of all marketing activities within Oracle Eloqua.

For your Salesforce integration, take some time to identify the external activities you want to leverage as part of closed-loop reporting. External activities can be attributed to Oracle Eloqua campaigns and shared with sales.

List the activities that you want to track using a table like the one below. After planning, create the external assets and activities. For more information, see [External assets and activities](#).

External asset type	External activity type
Trade show	Visited booth
	Viewed a demo
Webinar	Registered
	Attended
	Registered - did not attend

# Creating campaign response rules for closed-loop reporting

Campaign response rules define the campaign activities that are important to your organization and allow you to update the Salesforce campaign member status.

Campaign activities can include:

- Inbound activities, such as email opens, clickthroughs, and form submissions.
- External activities, such as trade shows and conferences.

Oracle Eloqua campaign responses are mapped to Salesforce campaign member status as determined by the response rule. Every hour (5 minutes after the hour) Oracle Eloqua evaluates campaign responses. When a response occurs, Oracle Eloqua records the response and later the response is sent to Salesforce.

Response rules are prioritized so that only the highest priority response is sent to Salesforce. This ensures that when an activity that is lower in priority occurs, it does not overwrite the campaign member status that is higher in priority. It also does not overwrite the action that caused the lead to be a response to a campaign in Oracle Eloqua.

Using a webinar campaign as an example, when someone initially clicks through on the invitation email, the member status changes to *Responded*. If they then complete the registration form, their member status changes to *Registered*. Registering for a webinar is a higher priority and is reflected in the Salesforce campaign member status.

For your closed-loop reporting implementation, consider the Oracle Eloqua inbound activities and external activities. Which activities matter most for determining conversion? What campaign member status should these activities map to?


Use a table like the one below to help you plan.

Priority	Response activity	Salesforce campaign member status
1	Tradeshow - Viewed a demo	Viewed a demo
2	Tradeshow - Visited booth	Visited booth
3	Webinar - Registered	Registered
4	Webinar - Attended	Attended
5	Form submit	Responded

### Before you begin:

- Set up your external activities before setting up response rules. See [Creating external assets and activities](#).
- Verify the campaign member status setup in Oracle Eloqua. You can import the status options from Salesforce, or create new ones in Oracle Eloqua. For more information, see [Creating campaign member status values](#).
- Oracle Eloqua evaluates campaign responses every hour (5 minutes after the hour). Campaign responses are recorded by Oracle Eloqua for 12 months after the campaign end date.
- We recommend that you do not create a response rule for the Email Sent activity. This would create a response whenever an email is sent to a contact.


### To create campaign response rules:

1. Click **Settings** .
2. Click **Response Rules** under *Database Setup*.
3. Click **Add Response Activity**. A new row appears in the *Default Campaign Response Rule Details* table.

Default Campaign Response Rules					
Default Campaign Response Rule Details					
<small>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.</small>					
	Priority	Response Activity	Salesforce Member Campaign Status	Salesforce Responded	Salesforce Default
	 	1 Response - Form Submit	Responded	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	 	2 External Activity : Showtrade Registered	Registered	<input checked="" type="checkbox"/>	<input type="checkbox"/>

 Add Response Activity

4. Map the Oracle Eloqua response to the corresponding CRM response using the **Response Activity** and **Member Campaign Status**.
5. Select the **Salesforce Responded** check box, then select the **Salesforce Default** check box for the campaign response rule that you want to set as the default. Enabling the *Salesforce Responded* setting ensures that the response is recorded in Salesforce as a response.
6. Repeat these steps until you have created all the response rules you want.

 **Note:** If you delete or modify a response rule, the change is applied the next time the response is sent to CRM.

7. Order the response rules in the table from highest priority to the lowest priority using the arrows table. Oracle Eloqua records all response activities but only sends over the highest priority response to CRM.

		Priority	Response Activity
✘	⬆ ⬇ ⬆	1	Response - Form St
✘	⬆ ⬇ ⬆	2	External Activity : St

8. Select **Use Program Builder for Campaign Associations** in the *Salesforce Integration* section.

**Salesforce Integration**

**Step 1: Define which Salesforce entities are created or updated.**

Set up integration rules if you wish to perform integration actions like creat 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 Salesforce entity.**

- Use Program Builder for Campaign Associations
- Use Program and Campaign Canvas for Campaign Associations
- Use Integration Rules for Campaign Associations

**Note:** This setting determines how responses are sent to Salesforce. We recommend using a program to update your CRM with campaign information. However, you organization might use another method. For more information on the program, see [Sending campaign data to Salesforce for closed-loop reporting](#).

9. Click **Save**.

# Configuring campaign revenue settings for closed-loop reporting

As part of closed-loop reporting with Salesforce, you must decide how to attribute revenue to Oracle Eloqua campaigns. The revenue attribution helps identify the marketing campaigns that help create opportunities and close a deal.

## To configure campaign measurement:

1. [Creating the Salesforce opportunity stages](#)
2. [Configuring the default revenue attribution method](#)

**Note:** Revenue attribution is for reporting purposes. This configuration does not impact how campaign and opportunity data is synchronized with Salesforce.

## Creating the Salesforce opportunity stages

As part of closed-loop reporting with Salesforce documented in this configuration guide, any opportunity created in Salesforce is included in Oracle Eloqua. Opportunity data includes:

- The opportunity itself (the deal in progress)
- The attributed campaign (whether this is the first campaign they responded to, the last

campaign, or a custom value)



- The opportunity's associated contact and that contact's activity

🌟 **Important:** To capture opportunity data from Salesforce, you need to create the Salesforce opportunity stages in Oracle Eloqua exactly as they appear in Salesforce. If the opportunity states set up in Oracle Eloqua do not match, the opportunity will not be imported from Salesforce.

### Before you begin:


- Create the Salesforce opportunity stages in Oracle Eloqua exactly as they appear in Salesforce. Opportunity stages are defined by your CRM administrator. They might include stages such as prospecting, developing, negotiation, closed (won), and closed (lost).
- The stages you create in Oracle Eloqua are active by default. You can deactivate them whenever necessary, but you cannot reactivate those you deactivate.
- You cannot change an opportunity stage in Oracle Eloqua after you save it.

### To create the Salesforce opportunity stages:

1. Navigate to **Orchestration** , then click **Campaigns**.
2. Click **Actions** , then click **Opportunity Settings**.
3. Click **Add New Opportunity Stage** in the *Opportunity Stage Details* area.
4. Enter the **Opportunity Stage Name** exactly as it appears in Salesforce.






5. Specify the **Probability of Closing** percentage. For example, for Closed Lost, this would be 0%.
6. If you want closed or lost opportunities to be flagged and excluded from the ROI calculations, select **Indicate Closed/Lost**.

 **Note:** This is the only setting you can change after you save the opportunity.

7. Click **Save and New** to set up additional stages.
8. Click **Save and Close** after you have created all of the required stages. The opportunities are displayed in the *Opportunity Stage Details* table.



### After you finish:

- If you want to change the closed or lost flag, click  next to the stage you want to change , then click **Edit**. Update the **Indicate Closed/Lost** setting.
- If you want to deactivate a stage, click  next to the stage you want to change and click **Inactivate Stage**. Warning: You cannot reactivate a stage after you deactivate it. This hides the stage but keeps a record for reporting purposes.
- Use  in the first column of the table to change the order of the opportunity stages. If you created the opportunity stages out of order, you can place them in order to match the sales cycle.

# Configuring the default revenue attribution method

After creating the opportunity sages in Oracle Eloqua, you must configure the default revenue attribution method. The revenue attribution method lets you identify the campaign responses to include in revenue attribution calculations and how to distribute attributed revenue across campaigns.

## To configure the default revenue attribution method for closed-loop reporting:

1. Navigate to **Orchestration** , then click **Campaigns**.
2. Click **Actions** , then click **Opportunity Settings**.
3. Click **Edit** in the *Default Revenue Attribution Method* area of the *Opportunity Settings* page.
4. Under *Which Campaign responses do you want to include in the revenue attribution?*, select an option to indicate which campaign responses you want to include in the revenue attribution. You can choose to include:
  - ALL Campaign responses from Lead Creation in Oracle Eloqua to Opportunity Close in the CRM. If this setting is selected, attribution will only include opportunities with a Closed Won status.
  - Campaign responses from Lead Creation in Oracle Eloqua to ONLY Opportunity Creation in the CRM. If this setting is selected, attribution will only include Open and Closed Won status opportunities.
5. Under *How do you want to distribute the revenue across the Campaigns?*, select an option to indicate how you want to distribute the revenue across campaigns. You can attribute the revenue in the following ways:
  - Evenly across all campaigns
  - To the first or last campaign responded to

- By specific percentages to the first, last, and other campaigns

6. Save your changes.

# Configuring auto synchs for closed-loop reporting

Auto synchs are scheduled imports to Oracle Eloqua. For closed-loop reporting with Salesforce, you must configure the following auto synchs:


- [Get campaigns](#)
- [Get opportunities](#)
- [Get Opportunities to associate opportunities with contacts](#)

## Preconfiguration: Creating folders

Auto synchs are organized into folders. If they don't already exist, you should create a Campaigns folder and a Opportunities folder for closed-loop reporting auto synchs.

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

### To create the required folder:

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

4. Click the **External Calls** tab in the left pane.
5. Under *All External Calls*, expand the **External Calls** folder.
6. Click the arrow beside the *Retrieve Data* folder, then click **Create New Folder**.
7. Enter **Campaigns** in the *Folder Name* box.
8. Add a folder description.
9. Click **Save and Close**.
10. Repeat this process to create the *Opportunities* folder in the same location.



## Creating the Get Campaigns auto synch

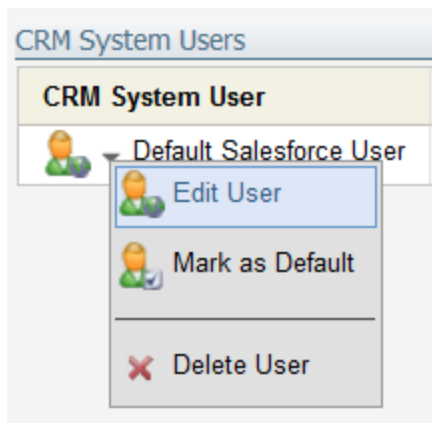
As part of closed-loop reporting, the Get Campaigns auto synch imports campaigns to Oracle Eloqua from Salesforce. The auto synch described here will only import those campaigns created by Oracle Eloqua. This means that campaigns created in Salesforce will not be imported. Campaigns in the systems are matched using the CRM campaign ID.

### **Before you begin:**

- Before setting up a new auto synch, check to see if it already exists. If the auto synch does exist, you can use the steps below to help configure the auto synch.
- To ensure that only campaigns created by Oracle Eloqua are importing, you will filter for campaigns created by the Oracle Eloqua integration user. To create this filter, you will require the user ID.


To get the Salesforce user ID from Oracle Eloqua

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Configuration** tab.
4. Click **Manage System CRM Users** in the *Configuration* menu.
5. In the *CRM System Users* area, click  next to *Default Salesforce User*, then click **Edit User**.

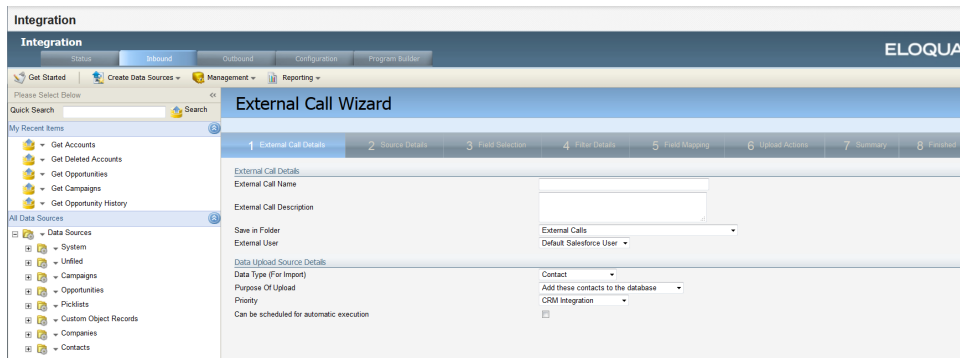


6. In the *Login Credentials* area of the *CRM System User* window, copy the text in the **User ID** box.
7. Click **Close**.
8. Click the *Inbound* tab to return to step 4 of the External Call Wizard.

### To configure the Get Campaigns auto synch:

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

4. Click **Create Data Source with External Call** in the *Create Data Sources* menu. The *External Call Wizard* opens displaying step 1 of 8.



5. Configure the settings in the *External Call Details* step.
- **External Call Name:** Get Campaigns
  - **External Call Description:** Get Salesforce Campaigns
  - **Save in Folder:** Retrieve Data - Campaigns
  - **External User:** Default Salesforce User
  - **Data Type (for Import):** Campaign
  - **Purpose of Upload:** Add New or Update Existing Campaigns
  - **Can be scheduled for automatic execution:** Enabled

6. Click **Next**.

7. Configure the settings in the *Source Details* step.
- **Action:** Retrieve
  - **Entity:** Campaign (Campaign)
  - **Retrieve 18 character IDs:** Enable if applicable (15 character IDs are typical when integrating with Oracle Eloqua)

**Note:** If you do enable this setting, you must ensure that all auto synchs are set up to retrieve 18 character IDs.

8. Click **Next**.
9. In the *Field Selection* step, select the Salesforce campaign fields that you want to populate in Oracle Eloqua when importing campaigns. For example, fields associated to campaign costs or fields that you might want to use for segmentation. You must select the Campaign ID as this will be used to uniquely match campaigns.

Salesforce Campaign Fields	Field Type	Select
Active	boolean	<input type="checkbox"/>
Actual Cost	currency	<input checked="" type="checkbox"/>
Budgeted Cost	currency	<input checked="" type="checkbox"/>
Campaign ID	id	<input checked="" type="checkbox"/>
Converted Leads	int	<input type="checkbox"/>
Created By ID	reference	<input type="checkbox"/>
Created Date	datetime	<input checked="" type="checkbox"/>

Salesforce Related Fields - CreatedBy

Salesforce Related Fields - LastModifiedBy

Salesforce Related Fields - Owner

10. Click **Next**.
11. In the *Filter Details* step, create a filter that looks for those campaigns created by Oracle Eloqua. Configure the first filter row as follows:
  - **Salesforce Campaign Fields:** Created By ID
  - **Operator:** Equals
  - **Expression:** The Salesforce ID of the Oracle Eloqua Integration User.



**Note:** Use additional filter criteria if you want only certain Salesforce campaigns to be pulled into Oracle Eloqua. For example, Record Type ID could be used to filter for only those campaigns related to specific business unites.

12. Click **Next**.
13. Configure the settings in the *Field Mapping* step to map the fields you selected in the *Field Selection* to the Oracle Eloqua campaign fields. The **Uniquely Match On** field should be set to *External Campaign ID*.

**Unique Match**

Uniquely Match On      Eloqua Campaign Field: External Campaign ID ▼

Perform a case-sensitive match

**Field Mappings**

Auto-Map Fields Advanced Options

Source Field	To	Target Field
Campaign: Actual Cost	➔	Actual Cost ▼
Campaign: Budgeted Cost	➔	Budgeted Cost ▼
Campaign: Campaign ID	➔	External Campaign ID ▼
Campaign: Created Date	➔	Created Date ▼

14. Click **Next** until you get to the *Summary* step.
15. In the *Summary* step, review the details. If no changes are required, select **Schedule automatic data imports using these settings**. Additional settings appear.
16. Set up the notification details and schedule the auto sync. We recommend:
  - **Auto Synch Name:** Get Campaigns
  - **Execute after:** No change

- **Send email on success:** Enabled
- **Send email on failure:** Enabled
- **Scheduling:** Leave as default

**Note:** Set a schedule that makes sense for your business process. The default schedule is runs every night at 10 PM EST.

Here is an example summary of the Get Campaigns auto synch.

Choice	Value
Name	Get Campaigns (ak) uploaded on 8/18/2017 10:36:41 AM EST.
Description	Get Campaigns (ak) uploaded by Alexa.Kalapaca on 8/18/2017 10:36:41 AM EST.
Data Upload Source	Get Campaigns (ak)
Upload Purpose	Add New or Update Existing Campaigns
Uniquely Match On	External Campaign ID

**Notification Email**  
Send a success or failure notification email to

**After Uploading**

Run once using these settings  
 Save these settings for re-use  
 Schedule automatic data imports using these settings

**Details**

Auto Synch Name:   
 Description:   
 Execute after:   
 Send email on success:   
 Send email on failure:

**Scheduling**

Auto-Synch Runs on Selected Days at 10pm EST?  Yes  No. A specified schedule governs Auto-Synch.

Selected Auto-Synch Days:

17. Click **Finish**.

To verify that the auto synch was created, on the **Inbound** tab, click **Management > Auto Synchs**.


# Creating the Get Opportunities auto synch

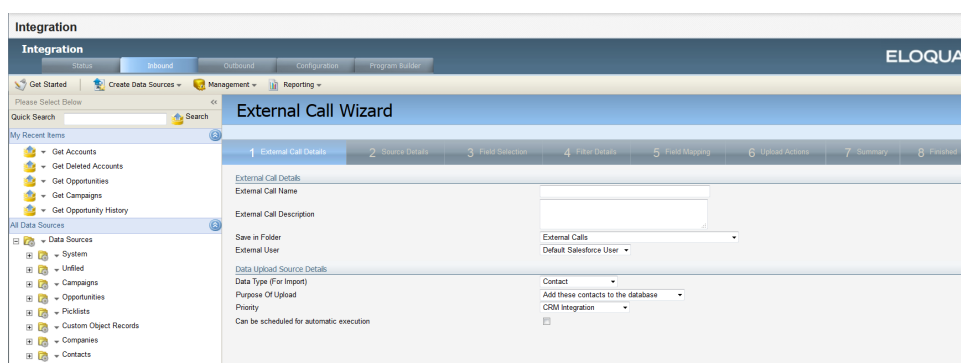
The Get Opportunities auto synch imports opportunities from Salesforce to Oracle Eloqua.

## Before you begin:

- Before setting up a new auto synch, check to see if it already exists. If the auto synch does exist, you can use the steps below to help configure the auto synch.

## To configure the Get Opportunities auto synch:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Inbound** tab.
4. Click **Create Data Source with External Call** in the *Create Data Sources* menu. The *External Call Wizard* opens displaying step 1 of 8.




5. Configure the settings in the *External Call Details* step.

- **External Call Name:** Get Opportunities
- **External Call Description:** Get SFDC Opportunities
- **Save in Folder:** Retrieve Data - Opportunities
- **External User:** Default Salesforce User
- **Data Type (For Import):** Opportunity
- **Purpose of Upload:** Add these opportunities to the database
- **Can be scheduled for automatic execution:** Enable

6. Configure the settings in the *Source Details* step.

- **Action:** Retrieve
- **Entity:** Opportunity
- **Retrieve 18 character IDs:** Enable if applicable (15 character IDs are typical when integrating with Oracle Eloqua)

 **Note:** If you do enable this setting, you must ensure that all auto synchs are set up to retrieve 18 character IDs.

7. Click **Next**.

8. Configure the settings in the *Field Selection* step. Select these Salesforce opportunity and account fields.

- Salesforce Opportunity Fields
  - **Amount**
  - **Close Date**
  - **Created Date**
  - **Name**

- **Opportunity ID**
- **Stage**
- **Salesforce Related Fields - Account**
  - **Account Name**
  - **Currency** (only if you use multiple currencies to record Salesforce Opp Amounts)
  - **Forecast to Close Date** (only if necessary; this field is not required for closed-loop reports)

9. Click **Next**.

10. In the *Filter Details* step, create following filter that looks for those opportunities created since the last import.

- **Salesforce Opportunity Fields:** Last Modified Date
- **Operator:** Greater Than or Equal
- **Expression:** Click  , then click **Add Date**. In the *Add Date* window, select **Use the start date of the last successful upload**

 **Note:** If you have specified a Record Type ID, create the filter for it.

11. Click **Next**.

12. Configure the settings in the *Field Mapping* step to map the fields you selected in the *Field Selection* to the Oracle Eloqua campaign fields. The **Uniquely Match On** field should be set to *Opportunity ID*.

- **Uniquely Match On:** Opportunity ID
- **Perform a case-sensitive match:** Enable



**Unique Match**

Uniquely Match On Eloqua Opportunity Field:

Perform a case-sensitive match

---

**Field Mappings**

 Auto-Map Fields  Advanced Options

Source Field	To	Target Field
Account: Account Name	➔	<input type="text" value="Account Name"/>
Opportunity: Amount	➔	<input type="text" value="Amount"/>
Opportunity: Close Date	➔	<input type="text" value="Close Date"/>
Opportunity: Created Date	➔	<input type="text" value="Created Date"/>
Opportunity: Name	➔	<input type="text" value="Opportunity Name"/>
Opportunity: Opportunity ID	➔	<input type="text" value="Opportunity ID"/>
Opportunity: Stage	➔	<input type="text" value="Stage"/>

13. In the *Summary* step, review the details. If no changes are required, select **Schedule automatic data imports using these settings**. Additional details appear.
14. Set up the notification details and schedule the auto sync. We recommend:
  - **Auto Synch Name:** Get Opportunities
  - **Execute after:** No change
  - **Send email on success:** Enabled
  - **Send email on failure:** Enabled
  - **Scheduling:** Configure a schedule that matches the frequency of your other Salesforce auto synchs. Oracle Eloqua only imports opportunities modified since the last successful import.

Here is an example completed summary step.

**Upload Settings**

Choice	Value
Name	Get Opportunities (ak) uploaded on 8/18/2017 10:53:53 AM EST.
Description	Get Opportunities (ak) uploaded by Alexa Kalapaca on 8/18/2017 10:53:53 AM EST.
Data Upload Source	Get Opportunities (ak)
Upload Purpose	Add these opportunities to the database
Uniquely Match On	Opportunity ID Perform a case-sensitive match

**Notification Email**

Send a success or failure notification email to

**After Uploading**

Run once using these settings  
 Save these settings for re-use  
 Schedule automatic data imports using these settings

**Details**

Auto Synch Name

Description

Execute after

Send email on success

Send email on failure

**Scheduling**

Auto-Synch Runs on Selected Days at 10pm EST?  Yes  No. A specified schedule governs Auto-Synch.

Selected Auto-Synch Days :

15. Click **Finish**.

To verify that the auto synch was created, on the **Inbound** tab, click **Management > Auto Synchs**.

## Creating the Get Opportunities auto synch to associate opportunities with contacts

For closed-loop reporting with Salesforce, you must create an auto synch to attribute revenue to a campaign. This is done by associating opportunities to Oracle Eloqua contacts. This auto synch runs after the [Get Opportunities auto synch](#) created earlier.

**Before you begin:**

- Before setting up a new auto synch, check to see if it already exists. If the auto synch does exist, you can use the steps below to help configure the auto synch.

## Choosing the right contact association

Before setting up this Get Opportunity auto synch, you must determine how you want to associate an opportunity to a contact. For example, should an opportunity be associated with a primary contact that the sale was closed with or with all contacts associated with the opportunity in the CRM? Should the opportunity be associated with accounts and all contacts on the account?

The opportunity association can be tied to contacts in one of the following ways:

- Link to primary contact:** Only the primary contact on the account is associated with the opportunity in Oracle Eloqua.
- Link to all contacts:** All contacts that are part of the opportunity in Salesforce are associated with the opportunity in Oracle Eloqua.
- Link to account contacts:** All contacts associated with the account are associated with the opportunity in Oracle Eloqua. This option requires that accounts with Salesforce account IDs have been imported into Oracle Eloqua and that contacts in Oracle Eloqua [are linked to accounts](#). All contacts in Oracle Eloqua that are associated with the account are associated with the opportunity.


The option you choose depends on your organization and how your business associates customers or prospects to opportunities. Some organizations associated an account to the opportunity with other associate the named contacts. For a short-term ROI approach, fewer contacts will provide better visibility into cause and effect. If Salesforce adoption or data management presents a challenge of inconsistent

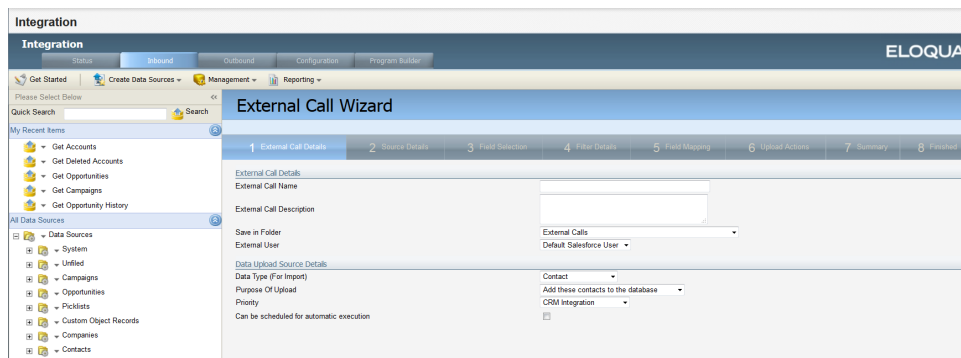


mapping of Contacts to Opportunities, option C may be the only choice. Whatever decision you make, it applies to all opportunities and closed-loop reports.

## Configuring the Get Opportunities auto synch for opportunity association

**To configure the Get Opportunities auto synch:**

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Inbound** tab.
4. Click **Create Data Source with External Call** in the *Create Data Sources* menu. The *External Call Wizard* opens displaying step 1 of 8.



5. Configure the settings in the *External Call Details* step.

Association Type	Settings
A. Link to primary contact	<p><b>External Call Name:</b> Get Opportunities and Link to Contacts</p> <p><b>External Call Description:</b> Get SFDC opportunities and link to contacts</p>
B. Link to all contacts	<p><b>Save in Folder:</b> Retrieve Data - Opportunities</p> <p><b>External User:</b> Default Salesforce User</p> <p><b>Data type (For Import):</b> Retrieve Data - Opportunity</p> <p><b>Purpose of Upload:</b> Link these opportunities to contacts</p> <p><b>Can be scheduled for automatic execution:</b> Enable</p>
C. Link to account contacts	<p><b>External Call Name:</b> Get Opportunities and Link to Accounts</p> <p><b>External Call Description:</b> Get SFDC opportunities and link to accounts</p> <p>The remaining settings are the same as for the <i>Link to primary contact</i> external call (A), described in the first row.</p>

6. Configure the settings in the *Source Details* step.


Association Type	Settings
A. Link to primary contact	<p><b>Action:</b> Retrieve</p> <p><b>Entity:</b> Opportunity Contact Role</p>
B. Link to all contacts	<p><b>Retrieve 18 character IDs:</b> Enable if applicable (15 character IDs are</p>

Association Type	Settings
	<p>typical when integrating with Oracle Eloqua)</p> <div style="background-color: #e1f5fe; padding: 10px; border-radius: 5px;"> <p><b>Note:</b> If you do enable this setting, you must ensure that all auto synchs are set up to retrieve 18 character IDs.</p> </div>
C. Link to account contacts	<p><b>Action:</b> Retrieve</p> <p><b>Entity:</b> Opportunity (Opportunity)</p> <p>The remaining settings are the same as for the <i>Link to primary contact</i> external call (A) above.</p>

7. Configure the settings in the *Field Selection* step. Select the following Salesforce contact role fields.

Association Type	Settings
A. Link to primary contact	<p><b>Contact ID</b></p> <p><b>Opportunity ID</b></p> <p><b>Primary</b></p>
B. Link to all contacts	<p><b>Contact ID</b></p> <p><b>Opportunity ID</b></p>
C. Link to account contacts	<p><b>Account ID</b></p> <p><b>Opportunity ID</b></p>

8. In the *Filter Details* step, create following filter that looks for those opportunities created since the last import.

- **Salesforce Opportunity Contact Role Fields:** Last Modified Date
- **Operator:** Greater Than or Equal
- **Expression:** Click , then click **Add Date**. In the *Add Date* window, select **Use the start date of the last successful upload**

**Note:** If you have specified a Record Type ID, include a filter for it.

9. Configure the settings in the *Field Mapping* step to map the fields you selected in the *Field Selection* to the Oracle Eloqua fields.

Association Type	Settings
A. Link to primary contact	<p><b>Uniquely Match on Eloqua Opportunity Field:</b> Opportunity ID</p> <p><b>Upload Source Field:</b> Opportunity Contact Role: Opportunity ID</p>
B. Link to all contacts	<p><b>Perform a case-sensitive match:</b> Enable</p>
C. Link to account contacts	<p><b>Uniquely Match on Eloqua Opportunity Field:</b> Opportunity ID</p> <p><b>Upload Source Field:</b> Opportunity: Opportunity ID</p> <p><b>Perform a case-sensitive match:</b> Enable</p>

10. Configure the settings in the *Upload Actions* step.

Association Type	Settings
A. Link to primary contact	<p><b>Source Field:</b> Opportunity Contact Role: Contact ID</p> <p><b>Field Name:</b> SFDC Contact ID (This is the Oracle Eloqua contact field)</p>

Association Type	Settings
B. Link to all contacts	<p>that stores the Salesforce contact ID.)</p> <p><b>Case sensitive match:</b> Enable</p> <p><b>Is Primary Contact:</b> Opportunity Contact Role: Primary</p>
C. Link to account contacts	<p><b>Source Field:</b> Opportunity Contact Role: AccountID</p> <p><b>Field Name:</b> SFDC Account ID (This is the Oracle Eloqua account field that stores the Salesforce account ID.)</p> <p><b>Case sensitive match:</b> Enable</p>

11. In the *Summary* step, review the details. If no changes are required, use the following settings.

Association Type	Settings
A. Link to primary contact	<p><b>Schedule automatic data imports using these settings:</b> Enable</p> <p><b>Auto Synch Name:</b> Get Opportunities and Link to Contacts</p>
B. Link to all contacts	<p><b>Description:</b> Get SFDC opportunities and link to contacts</p> <p><b>Execute after:</b> Get Opportunities</p> <p><b>Send email on success:</b> Optional</p> <p><b>Send email on failure:</b> Enable</p>
C. Link to account contacts	<p><b>Auto Synch Name:</b> Get Opportunities and Link to Accounts</p> <p><b>Description:</b> Get SFDC opportunities and link to accounts</p> <p><b>Execute after:</b> Get Opportunities</p>

Association Type	Settings
	<b>Send email on success:</b> Optional
	<b>Send email on failure:</b> Enable

12. Configure a schedule that matches the frequency of your [Get Opportunities](#) auto synch.
13. Click **Finish**.

To verify that the auto synch was created, on the **Inbound** tab, click **Management > Auto Synchs**.


## Disabling auto synchs for closed-loop reporting

If you do not want to import Salesforce campaigns or opportunities into Oracle Eloqua until after you have performed testing, disable the auto synchs.

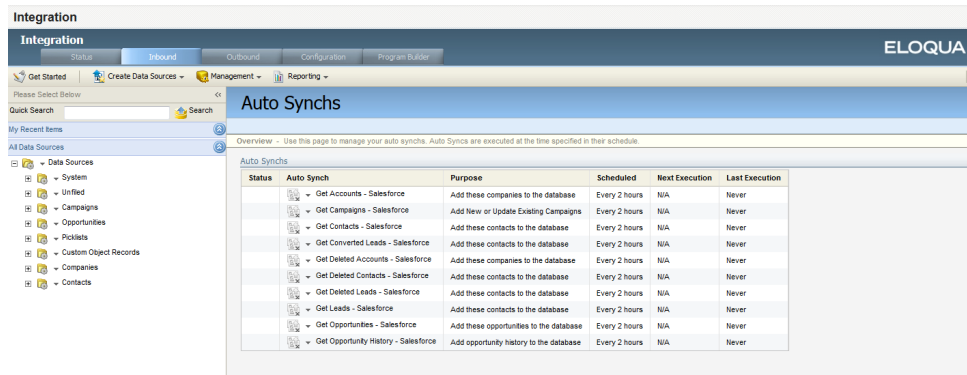
### Before you begin:

- Create the [auto synchs for closed-loop reporting](#).

### To disable the campaign and opportunity auto synchs:

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

4. Click **Management > Auto Synchs**.



5. For each of the closed-loop reporting auto synchs in the list, select **Disable Auto Synch** from the drop-down list.

# Configuring external calls for closed-loop reporting

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

The initial integration with Salesforce [included external calls to synchronize leads and contacts](#). Now you must configure the external calls required for closed-loop reporting:

- Create Campaign
- Update Campaign
- Contact Campaign Association
- Lead Campaign Association

These external calls will be triggered by internal activities that will be enabled at the end of the close-loop reporting configuration. See [Initializing closed-loop reporting](#) for more information.

## **Before you begin:**

- Create any necessary campaign fields in Oracle Eloqua. See [Creating campaign fields](#) for more information.
- The external calls used for campaign association may have been setup after you ran the [CRM Integration Wizard](#). If is the case, you can use the steps below to configure those calls instead of creating new ones.




# Preconfiguration: Creating folders

External calls are organized into folders. If they don't already exist, you should create a Campaigns folder for the closed-loop reporting external calls.

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


## To create the required folders:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Outbound** tab.
4. Click the **External Calls** tab in the left pane.
5. Under *All External Calls*, expand the **External Calls** folder.
6. Click the arrow beside the *Send Data* folder, then click **Create New Folder**.
7. Enter **Campaigns** in the *Folder Name* box.
8. Add a folder description.
9. Click **Save and Close**.

# Creating external calls

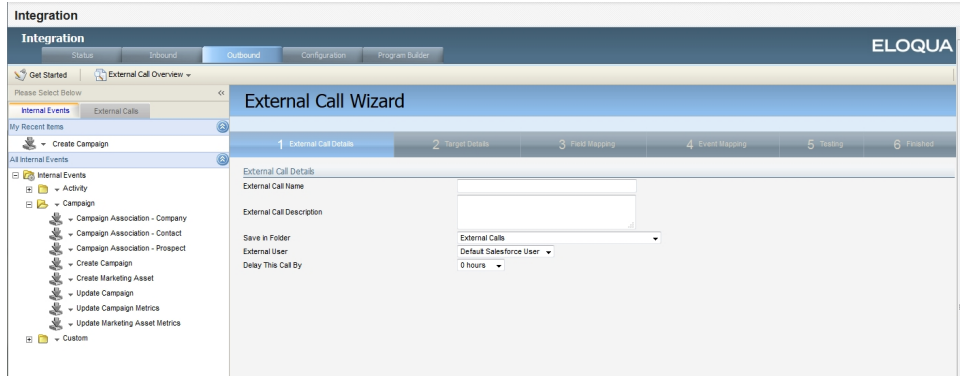
Before creating an external call, check to see if it already exists. If the external call does exist, you can use the steps below to help configure it.

## To create the external calls for closed-loop reporting:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Outbound** tab.
4. On the *Internal Events* tab, expand the **Campaign** folder.
5. Click the arrow next to the internal event, and select **Create New External Call**. Refer to the following table for the internal events to use.

External call	Internal event
Create Campaign	<b>Create Campaign</b>
Update Campaign	<b>Update Campaign</b>
Contact Campaign Association	<b>Campaign Association - Contact</b>
Lead Campaign Association	<b>Campaign Association - Contact</b>

The *External Call Wizard* opens.

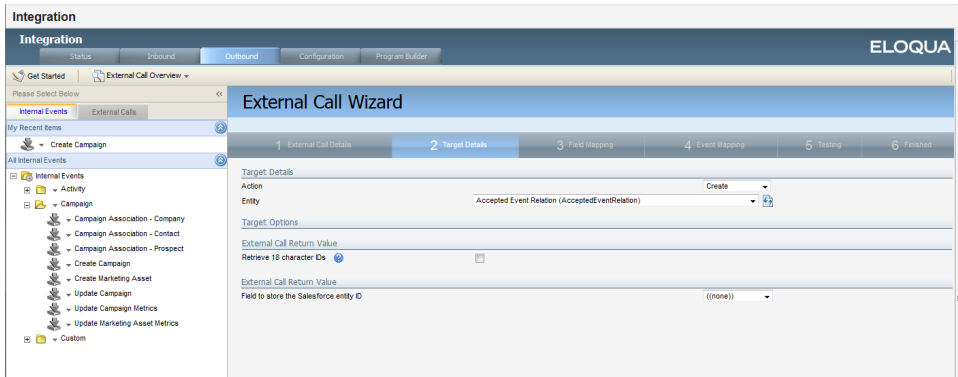


6. Configure the *External Call Details* settings for each of the four external calls as follows:

External call	Configuration
Create Campaign	<p><b>External Call Name:</b> Create Campaign</p> <p><b>External Call Description:</b> (Optional)</p> <p><b>Save in Folder:</b> Send Data - Campaigns</p> <p><b>External User:</b> Default Salesforce User</p> <p><b>Delay This Call By:</b> 0</p> <p>This delay setting means that there will be no time delay to start the activity after it is triggered.</p>
Update Campaign	<p><b>External Call Name:</b> Update Campaign</p> <p>The remaining settings are the same as for the Create Campaign external call.</p>
Contact Campaign Association	<p><b>External Call Name:</b> Contact Campaign Association</p> <p>The remaining settings are the same as for the Create Campaign external call.</p>
Lead Campaign	<p><b>External Call Name:</b> Lead Campaign Association</p>

External call	Configuration
Association	The remaining settings are the same as for the Create Campaign external call.

7. Click **Next**. The *Target Details* settings are displayed.

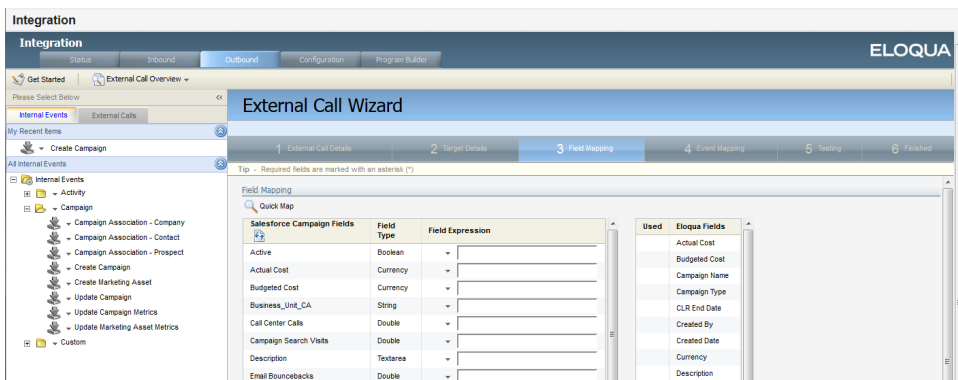


8. Configure the *Target Details* settings as follows:

External call	Configuration
Create Campaign	<p><b>Action:</b> Create</p> <p><b>Entity:</b> Campaign</p> <p><b>Retrieve 18 character IDs:</b> Enable if applicable (15 character IDs are typical when integrating with Salesforce)</p> <div style="background-color: #e1f5fe; padding: 10px; margin: 10px 0;"> <p><b>Note:</b> If you do enable this setting, you must ensure that all external activities are set up to retrieve 18 character IDs.</p> </div> <p><b>Field to store the Salesforce entity ID:</b> System Defined</p>

External call	Configuration
Update Campaign	<p><b>Action:</b> Update</p> <p>The remaining settings are the same as for the Create Campaign external call.</p>
Contact Campaign Association	<p><b>Action:</b> Create</p> <p><b>Entity:</b> Campaign Member</p> <p>The remaining settings are the same as for the Create Campaign external call.</p>
Lead Campaign Association	<p>Configure these settings the same as the Contact Campaign Association external call.</p>

9. Click **Next**. The *Field Mapping* settings are displayed.



10. Configure the mappings between Salesforce and Oracle Eloqua fields by dragging and dropping Oracle Eloqua fields into the *Field Expression* column. The following table provides details for configuring mappings for each external call.

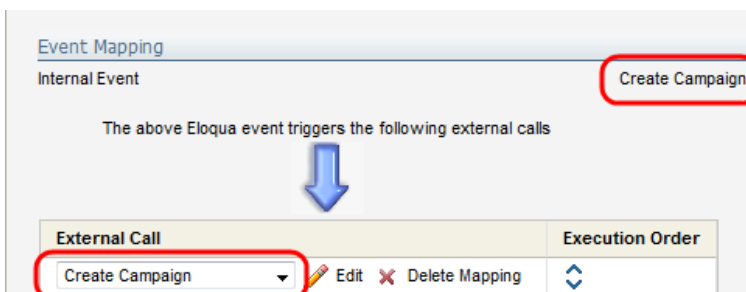
External call	Mapping configuration details
Create Campaign	<ul style="list-style-type: none"> <li>For the Salesforce <i>Active</i> field, set the value to 1. This will set the</li> </ul>

External call	Mapping configuration details
	<p>campaign as active in Salesforce when the campaign information is sent over.</p> <ul style="list-style-type: none"> <li>Map Oracle Eloqua fields to Salesforce to meet your specific requirements.</li> </ul>
Update Campaign	<ul style="list-style-type: none"> <li>Map the Oracle Eloqua <i>External Campaign ID</i> field to the Salesforce <i>Campaign ID</i> field.</li> <li>Map other Oracle Eloqua fields to Salesforce to meet your specific requirements.</li> </ul>
Contact Campaign Association	<ul style="list-style-type: none"> <li>Map the Oracle Eloqua <i>Campaign ID</i> to Salesforce <i>Campaign ID</i>.</li> <li>Map the Oracle Eloqua <i>SFDC Contact ID</i> to Salesforce <i>Contact ID</i>.</li> <li>Map the Oracle Eloqua <i>Status</i> to Salesforce <i>Status</i>.</li> </ul>
Lead Campaign Association	<p>Map only the following Oracle Eloqua fields to Salesforce:</p> <ul style="list-style-type: none"> <li>Map the Oracle Eloqua <i>Campaign ID</i> to Salesforce <i>Campaign ID</i>.</li> <li>Map the Oracle Eloqua <i>SFDCLeadID</i> to Salesforce <i>Lead ID</i>.</li> <li>Map the Oracle Eloqua <i>Status</i> to Salesforce <i>Status</i>.</li> </ul>


11. Click **Next**. The *Event Mapping* page opens.

12. Confirm that the internal event triggers external call you are creating. Click **Next**.

The following image shows that the Create Campaign internal event is triggering the Create Campaign external call.



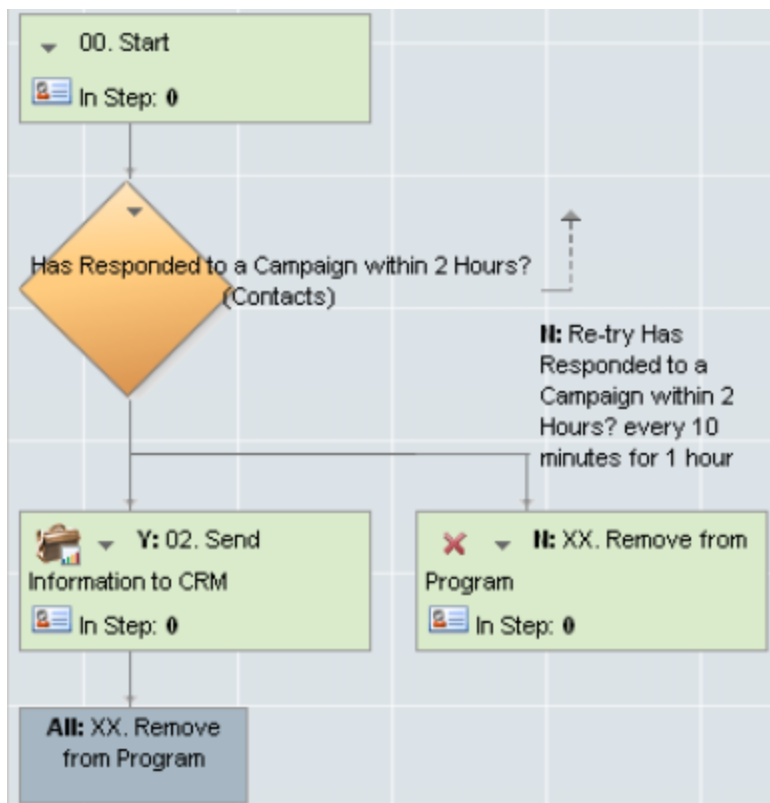
13. The *Testing* step is optional.

- a. Click  to select an existing Oracle Eloqua campaign to test the external call.
- b. Click **Prepare for Test** to ensure there are no errors.
- c. Click **Execute Test** when there are no errors to be resolved.
- d. Use the *Testing Data* table to confirm that the Oracle Eloqua data was sent to Salesforce.

14. Click **Finish**.

# Sending campaign data to Salesforce for closed-loop reporting

The *SYSTEM - Send Campaign information to CRM* program sends campaign data to Salesforce. It is run after the *SYSTEM - CRM Update* program. This program creates campaign memberships in Salesforce based on the campaign responses that occurred in Oracle Eloqua.



The program works in the following way:

- Oracle Eloqua evaluates responses every hour (5 minutes after the hour). Responses are captured and recorded against the appropriate Oracle Eloqua campaign based on the response rule setup.



- After contacts flow through the [SYSTEM - Update CRM program](#), they enter into the [SYSTEM - Send Campaign information to CRM](#) program. A decision rule looks for contacts that have responded within the last 2 hours.
- If contacts meet the filter criteria, the action step sends all new responses over to Salesforce using the [Campaign Association - Contact](#) internal event.
- If contacts do not meet the filter criteria, the decision rule is retried for 2 hours to ensure a response is not missed.

## Before you begin

- Set up response rules. See [Creating campaign response rules for closed-loop reporting](#).
- Set up the external calls. See [Configuring external calls for closed-loop reporting](#)


## Filtering contacts that responded to campaigns

The [SYSTEM - Send Campaign information to CRM](#) program requires a shared contact filter that includes contacts that responded to a campaign within the last 2 hours.

The screenshot shows a configuration window titled "Add New Shared Filter" for a filter named "SYSTEM-Responded to Campaign in Last 2 Hours". The filter is currently active, as indicated by a red status icon and the text "Responded to Any Campaign" and "Contacts who have responded to any campaign at least 1 time on 2/10/2017". The configuration details are as follows:

Field	Value
# of Responses	at least 1 time(s)
Timeframe	within the last 2 Hours
Campaign Fields	All

**To create a shared filter to include contacts that responded to a campaign:**

1. Navigate to **Audience**  > **Tools**, then click **Shared Filters**.
2. Click **New**.
3. Double-click **Untitled Shared Filter**, then enter the filter name: SYSTEM - Responded to Campaign in Last 2 Hours.
4. Add the **Responded to Any Campaign** filter criteria.
5. Modify the criteria to include contacts who have responded to any campaign at least once within the last 2 hours. Keep the default setting (**All**) for the *Campaign Fields* option.


**Contacts who have responded to any campaign...**

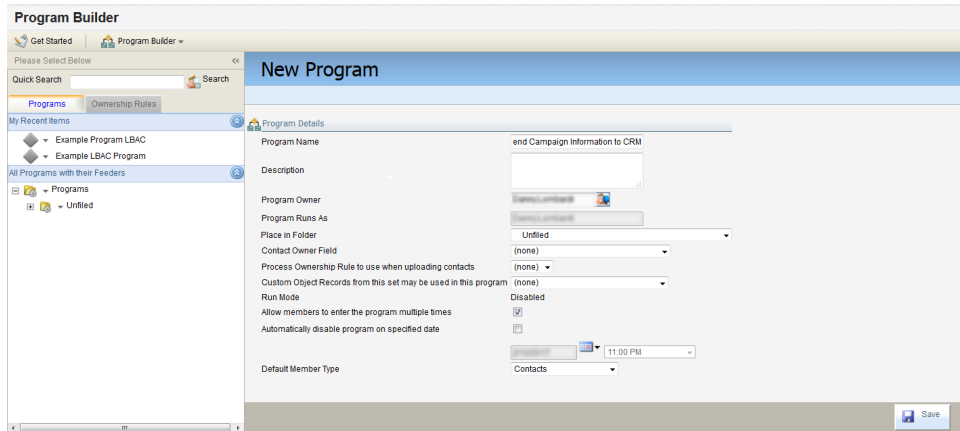
# of Responses	at least	▼	1	time(s)
Timeframe	within the last	▼	2	Hours ▼
Campaign Fields	All	▲▼		

6. Click **Save**.

## Creating the send campaign information program


**To build the *SYSTEM - Send Campaign Information to CRM* program for closed-loop reporting:**

1. Navigate to **Orchestration** , then **Tools**, then click **Program Builder**.
2. Click **Create New Program** from the *Program Builder* menu.
3. Enter the program name, **SYSTEM – Send Campaign Information to CRM**, and select **Contacts** as the *Default Member Type*.

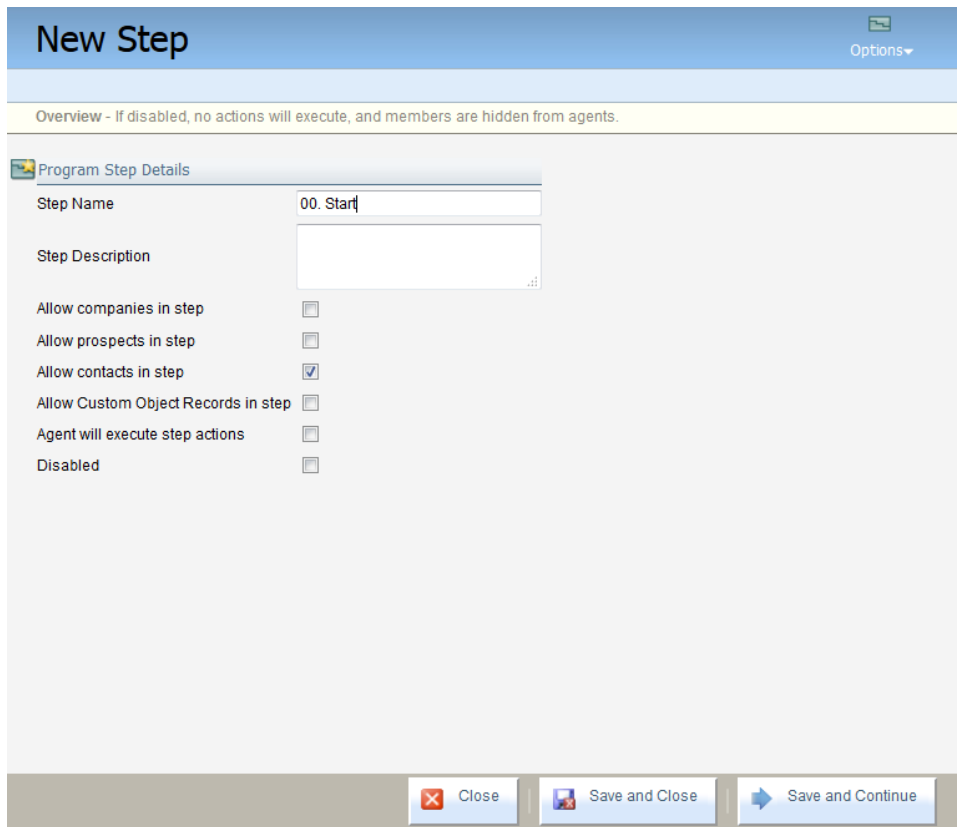


4. Click **Save**.

### Configure the program entry step and decision step:

5. Click **Edit Program Flow**.
6. Click  to add the first step in the program.

7. Name the first step **00. Start**, then click **Save and Close**.



**New Step** Options

Overview - If disabled, no actions will execute, and members are hidden from agents.

**Program Step Details**

Step Name: 00. Start

Step Description:

Allow companies in step

Allow prospects in step


Allow contacts in step

Allow Custom Object Records in step

Agent will execute step actions

Disabled

Close Save and Close Save and Continue

8. Click  beside the step name, then click **Edit Step Path**.
9. Select **Send to New Decision Rule**, then click **Continue**.
10. Name the decision rule **Has Responded to any Campaign in the Last 2 Hours?**

11. Select **Data** in the *Type* list, then select **Contact is Contact Filter** in the *Decision Rule* list.

**Edit Decision Rule**

Decision Rule Details

Decision Rule Name: Has Responded to any Campaign in the Last 2 Hours?

Evaluate: Immediately


Entity Type: Contacts

Type: Data


Decision Rule:

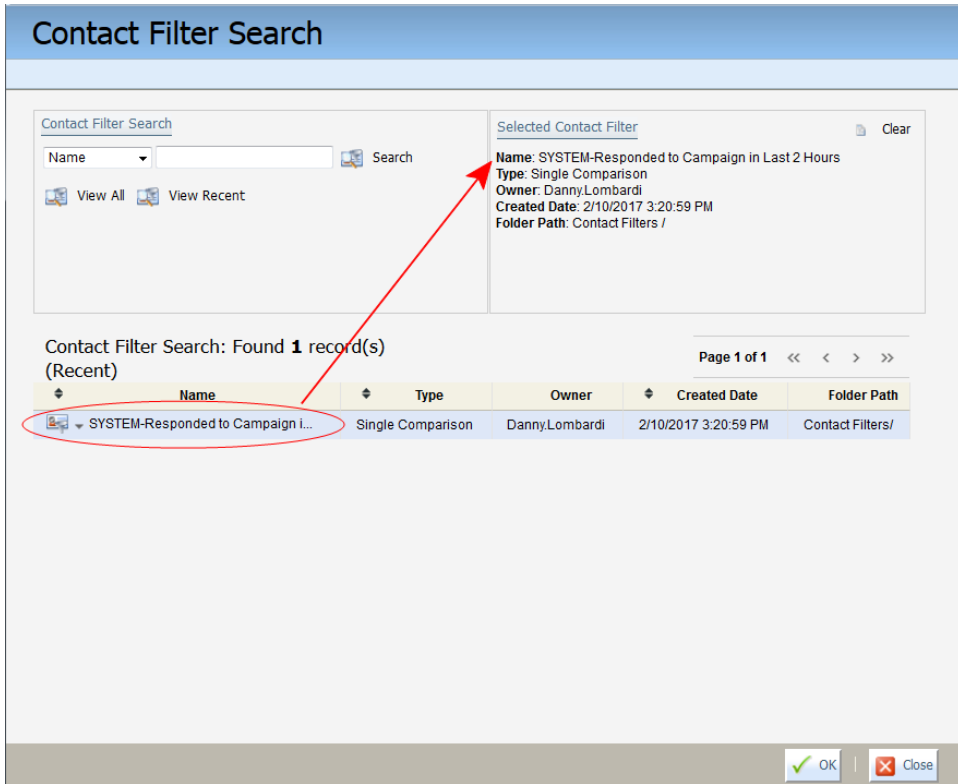
- Access to Contact restricted by Security Label
- Contact Field Comparison
- Contact Field has Value
- Contact has Membership in Another Program
- Contact is in Contact Filter**
- Contact is in Contact Group
- Contact's Linked Company Field Comparison
- Does Contact have "bounceback" status?
- Does Contact have "unsubscribe" status?

Decision Rule Parameters

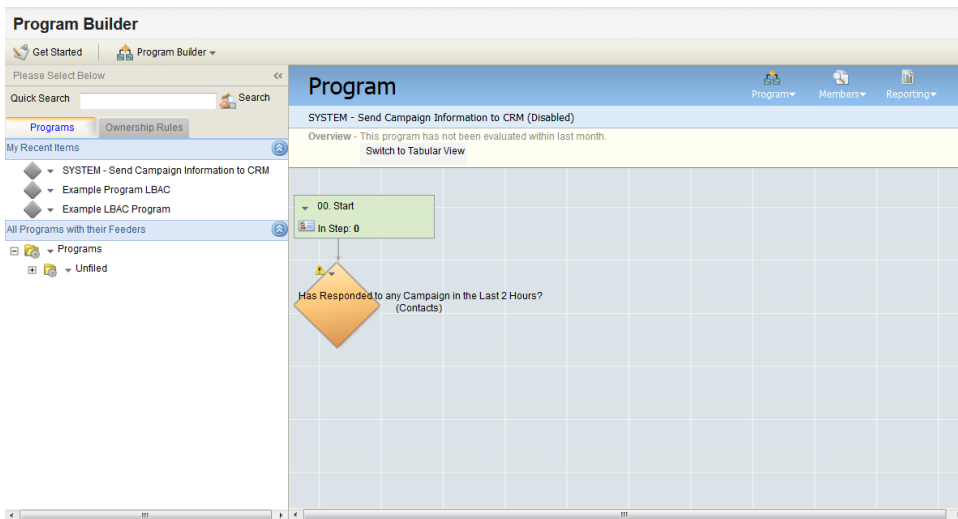
Filter: 

Close | Save and Close



12. Click  in the *Decision Rule Parameters* area, then select the shared filter you created earlier: SYSTEM – Responded to Campaigns within the last 2 hours. Click **OK**. See [Filtering contacts that responded to campaigns](#).




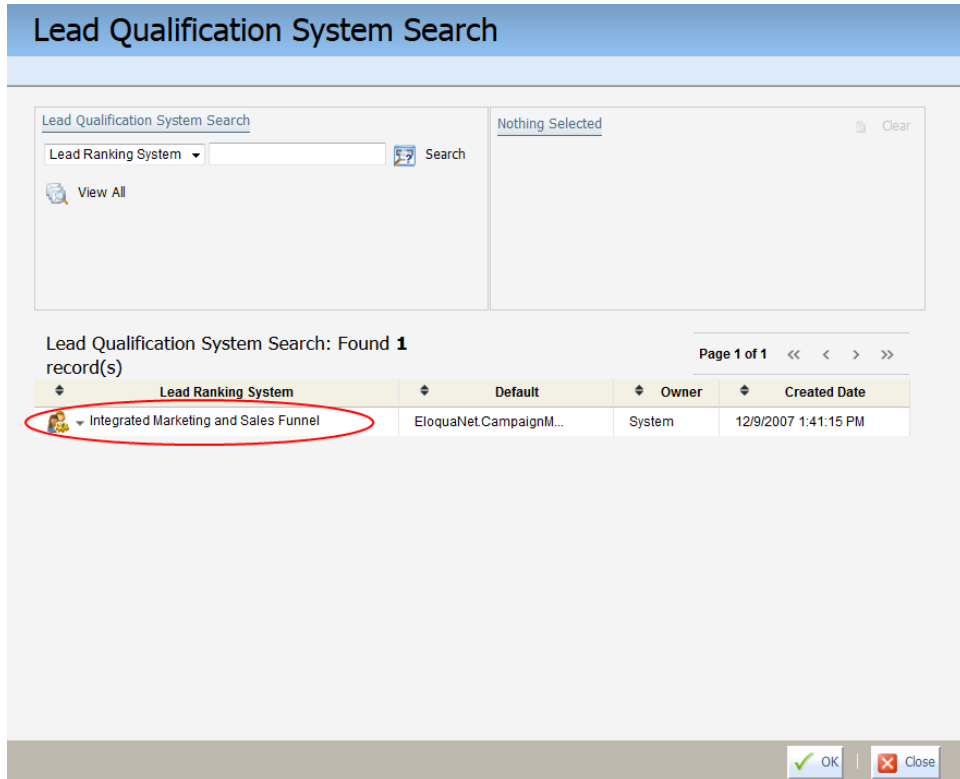
13. Click **Save and Close**. The program is updated with the new step.



**Configure the yes path:**



14. Click  on the *Has Responded to any Campaign in the Last 2 Hours* decision rule, then click **Edit Decision Yes Path**.
15. Select **Send to a new step**.
16. Click **Continue**.
17. Name the step **01. Send Campaign Information to CRM**, then click **Save and Continue**.
18. Name the action **01. Send Campaign Information to CRM**, then select **Send Campaign Information to CRM System** in the *Action* list.
19. Click  in the *Action Parameters area*, then select the default lead stage system identified in the search results.

 **Note:** You must select a Lead Stage System in order to enable the program.



20. Click **OK**, then click **Save and Close**.

### Configure the no path:

21. Click  on the *Has Responded to any Campaign in the Last 2 Hours* decision rule, then click **Edit Decision No Path**.
22. Select **Send to New Step**.
23. Check **Re-try decision rule every**, then select **10 minutes** for **1 hour** starting with **Has Responded to a Campaign within 2 Hours**.
24. Click **Continue**.
25. Click  on *Step 01*, then click **Edit Step Path**.
26. Select the option **Send to New Step**.



27. Name the action **XX. Remove from Program**, then select the **Remove from Program** action.

28. Click **Save and Close**.

### **Configure the priority mode:**

29. It is recommended you enable the program in priority mode.

- a. Click **Enable Program** from the *Program* menu.
- b. Select **Enable** below *Priority Mode*.




# Updating the CRM update program for closed-loop reporting

After [creating the program to send campaign data to Salesforce](#), you must update the *SYSTEM - CRM Update* program. The *SYSTEM - CRM Update* program is the primary program used to integrate Salesforce with Oracle Eloqua. You must update the program to direct contacts from the *SYSTEM - CRM Update* program to the *SYSTEM - Send Campaign information to CRM*.

## Before you begin:

- Create the *SYSTEM - Send Campaign information to CRM* program. See [Sending campaign data to Salesforce for closed-loop reporting](#).

## To update the CRM Update program for closed-loop reporting:

1. Navigate to **Orchestration** , then **Tools**, then click **Program Builder**.
2. Locate and click the CRM update program. You can use quick search or the folder pane.
3. Click **Disable Program** in the *Program* menu on the right.
4. Click  on the *XX. Remove from Program* step, then click **Edit Step Default Action**.
5. Change the action from **Remove from Program** to **Move to Another Program**.
6. Click  under *Action Parameters*. In the pop up window, search for the *SYSTEM - Send Campaign information to CRM* program.

7. Select the first step in the program, then click **Close**.

Process Step Search

Process Step Search

Program Name: Send Search

View All

Selected Process Step

Program Name: SYSTEM: Send Campaign Info to CRM  
Step Name: 00. Start Step  
Disabled: No

Process Step Search: Found 9 records Page 1 of 1

Program Name	Step Name	Description	Disabled
zzSYSTEM - Send Campaign1 Information to CRM	00. Start		No
zzSYSTEM - Send Campaign1 Information to CRM	01. Send Campaign Info to CRM		No
zzSYSTEM - Send Campaign1 Information to CRM	XX. Remove from Program		No
SYSTEM: Send Campaign Info to CRM	00. Start Step		No
SYSTEM: Send Campaign Info to CRM	01. Send Campaign Info to CRM		No


8. Save your changes.
9. Click **Enable Program** in the *Program* menu.
10. Click **Enable** for the mode you want to use.
11. Click **Close**.

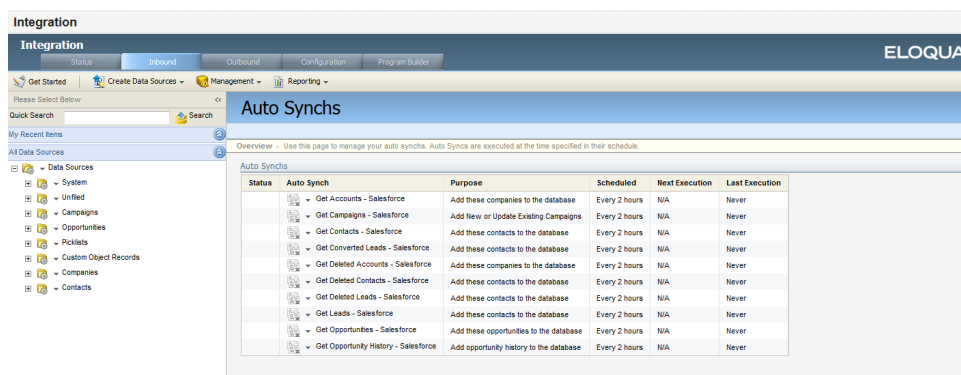
# Initializing closed-loop reporting

The final step in closed-loop reporting is enabling auto synchs and internal events.

## Enabling the auto synchs

To enable auto synchs:

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




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

5. For each of the closed-loop reporting auto synchs in the list, select **Enable Auto Synch** from the drop-down list.

# Enabling the internal events


Internal events trigger the external activities created earlier. For more information, see [Configuring external calls for closed-loop reporting](#).

## To enable the internal events:

1. Click **Settings** .
2. Click **Integration** under *Platform Extensions*.
3. Click the **Outbound** tab.
4. On the *Internal Events* tab, expand the **Campaign** folder.
5. For each of the following events in the folder, click **Enable Internal Event** from the drop-down list:
  - Campaign Association - Contact
  - Create Campaign
  - Update Campaign

# Validating auto synch processing

## 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 the history of the auto synch, click **Auto Synch History** from the drop-down list:
  - Campaign Association - Contact
  - Create Campaign
  - Update Campaign


## Configuring system notifications about errors

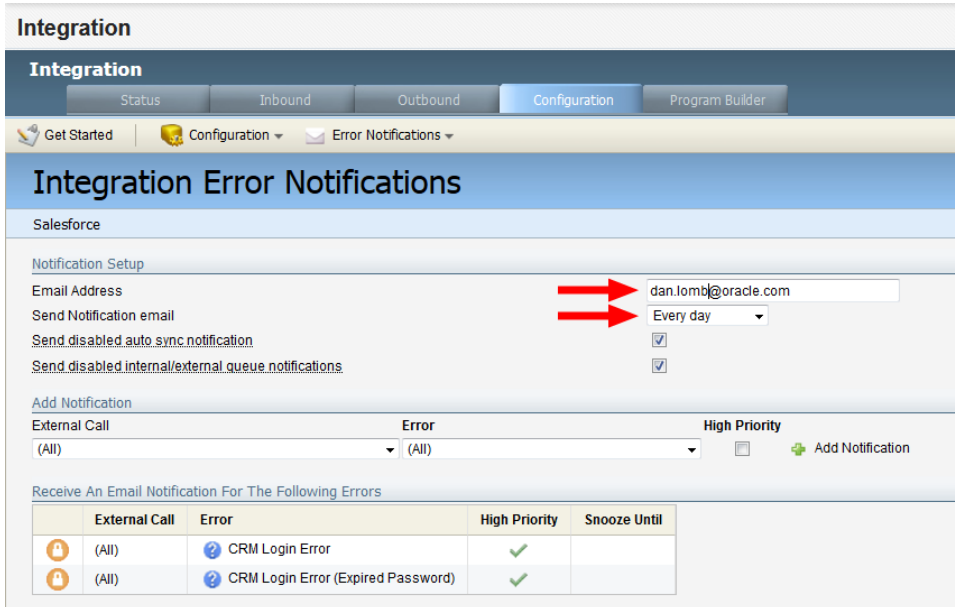
With both the integration auto synchs and the internal and external queues re-enabled, you can now set up error notifications.

Errors can cause issues with the synchronization between Oracle Eloqua and Salesforce. Configure error notifications to send alerts when specified errors occur. Determine who on your team should receive these notifications on an on-going basis. For closed-loop reporting, these are the recommended errors to trigger notification:

External Call: Create Campaign	Error: (All)
External Call: Update Campaign	Error: (All)
External Call: Contact Campaign Association	Error: (All)
External Call: Lead Campaign Association	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  dan.lomb@oracle.com

Send Notification email  Every day

Send disabled auto sync notification

Send disabled internal/external queue notifications

**Add Notification**

External Call (All) Error (All) High Priority  Add Notification

Receive An Email Notification For The Following Errors

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**.

## Closed-loop reports

After initializing closed-loop reporting, you can use the following Insight reports to evaluate campaign performance:

- **Campaign Cost Metrics:** Shows the results of campaign spend in terms of cost per response, cost per lead, and cost per opportunity.
- **Campaign Revenue Analysis:** Shows the attributed revenue results, resulting ROI and cost metrics for each campaign.
- **Opportunity Overview:** Shows a list of the opportunities and deals.

For more information on Insight, see [Insight](#).