

iRaiser Fundraising for Salesforce

Administrator Manual

Table of contents

Table of contents	2
General	3
Get Started	3
Step 1: Install package	3
Step 2: Configure External Credential	3
Step 3: Configure iRaiser settings	3
Basic configuration	4
Permission set configuration	4
Step 4: Configure Site	4
Step 5: Assign Admin Permissions	6
Monitoring and Management	6
Retry	6
Mapping Webhook Notifications	7
Cleanup	9
Need help?	10

General

The iRaiser Fundraising for Salesforce package allows the Salesforce admin to configure a webhook-based integration of iRaiser data into Salesforce. Incoming notifications are stored in Salesforce, enriched through the iRaiser API and handed to a Flow that you control, so all mapping and business logic stays in the hands of the admin.

Get Started

Step 1: Install package

Search the AppExchange for "iRaiser for Salesforce", click the listing with the iRaiser logo and Install the package.

Step 2: Configure External Credential

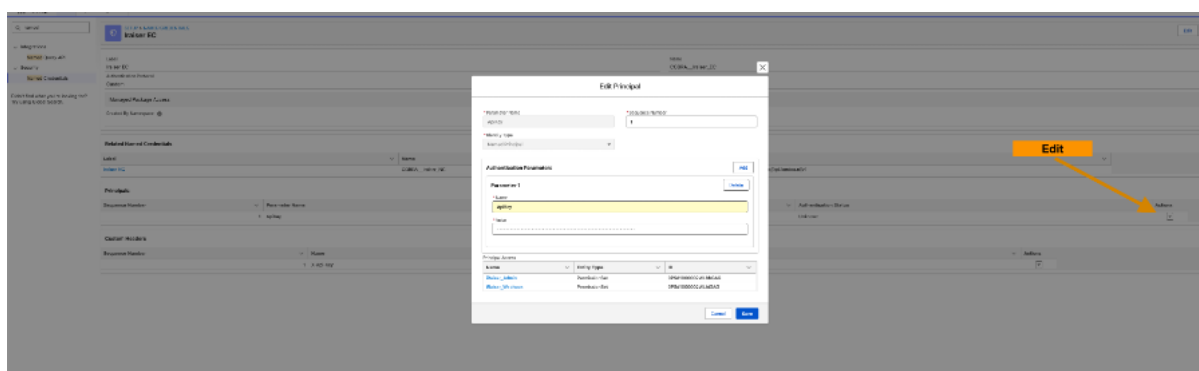
Navigate to **Setup** → **Security** → **Named Credentials** and find **iRaiser NC**. In the column **External Credential**, click **iRaiser EC**.



Named Credentials — open the external credential iRaiser EC.

In the next screen, click **Edit**, then under **Principals** click **Add** in the pop-up.

Fill in **ApiKey** as name and, for value, paste your iRaiser API key. This API key can be obtained from iRaiser or from the organisation you're implementing the package for.



Add the ApiKey principal to the external credential.

Step 3: Configure iRaiser settings

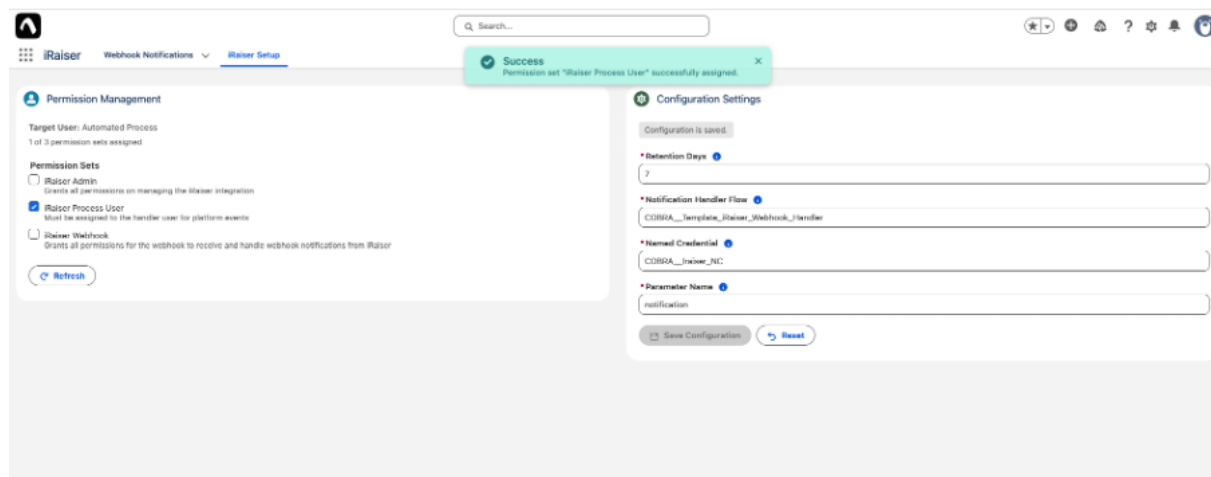
Navigate to the **iRaiser** app in the App Launcher. In the app, open the tab **iRaiser Setup**.

Basic configuration

The form on the right-hand side configures the behaviour of the iRaiser integration. It is important to save these settings at least once; most of the time the defaults are fine.

Field	Description
Retention Days	How long processed COBRA__Webhook_Notification__c records are retained, in days. After this time the cleanup batch removes the records, if scheduled.
Notification Handler Flow	The flow that handles incoming webhook notifications from iRaiser. Defaults to the packaged template.
Named Credential	The named credential used to contact the iRaiser API. Defaults to the named credential in the package.
Parameter Name	Name of the parameter that accepts the notification data in the flow configured in Notification Handler Flow .

When saving, a confirmation message should appear.



iRaiser Setup — confirmation after saving the settings.

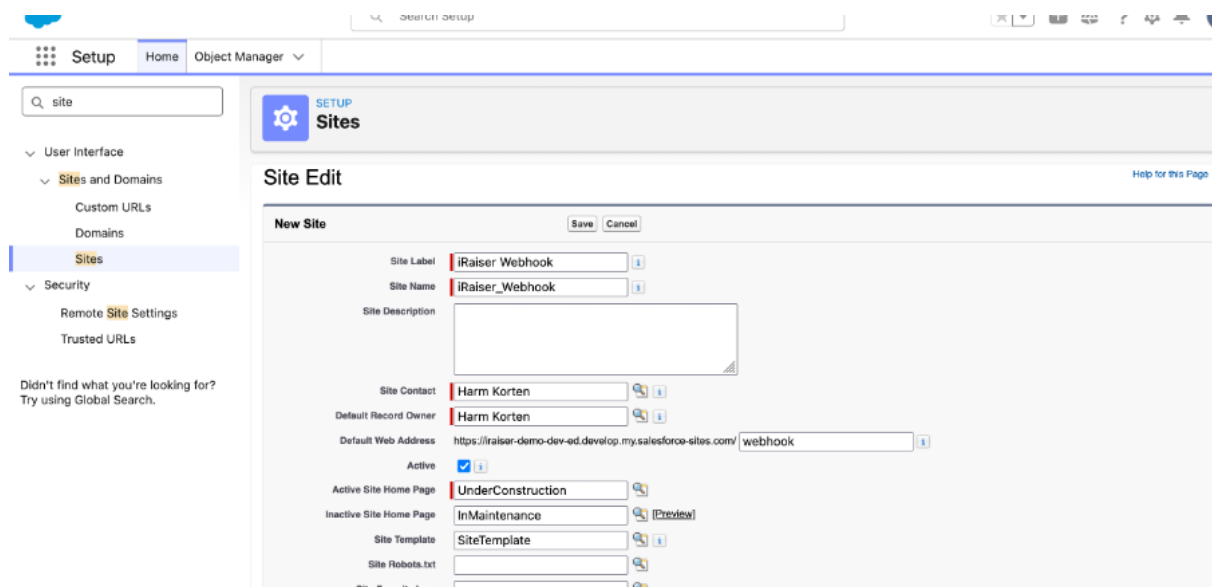
Permission set configuration

On the left-hand side, permission sets can be assigned to the Automated Process user. The minimum it needs is the **iRaiser Process User** permission set, which allows the Automated Process user to access the External Credential.

Step 4: Configure Site

Navigate to **Setup → User Interface → Sites and Domains → Sites**.

Click **New**, fill in the site form, make sure to check the **Active** box, and press **Save**.

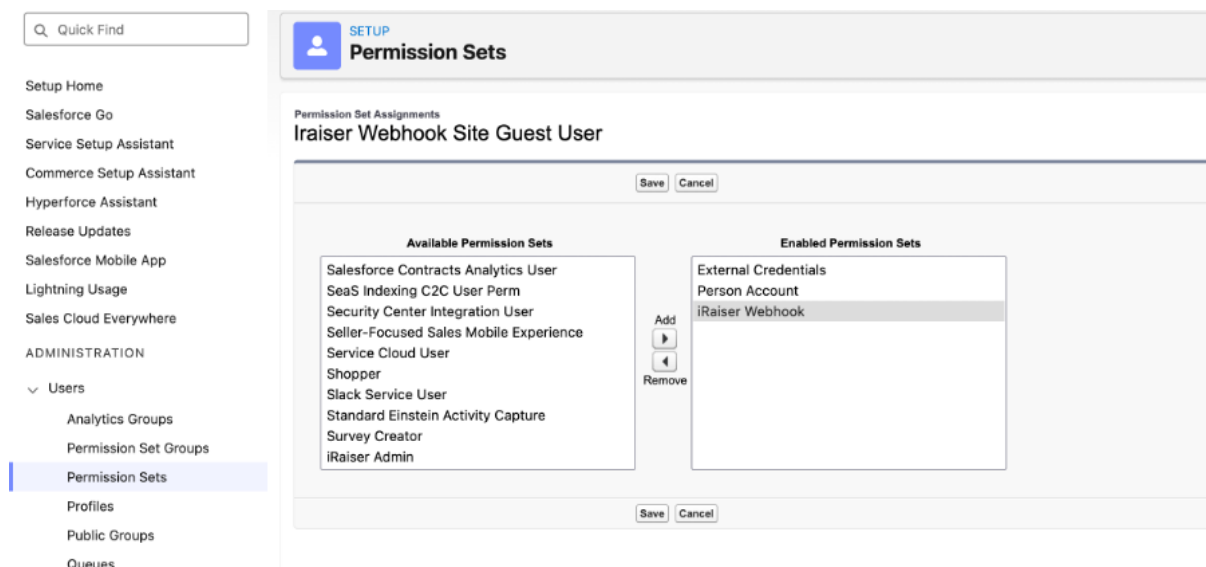


Site configuration for the webhook endpoint.

After configuring the site, it should appear in the list. The URL configured for the site should be provided to the iRaiser team, to be configured on the iRaiser backend as webhook URL. In the example from the screenshot above, the URL would be:

<https://iraider-demo-dev-ed.develop.my.salesforce-sites.com/webhook/services/apexrest/COBRA/webhook>

Click the site and click the **Public Access Settings** button. This brings you to the site's guest profile page. Click **Assigned Users**, click the Guest user and assign the **iRaiser Webhook** permission set.

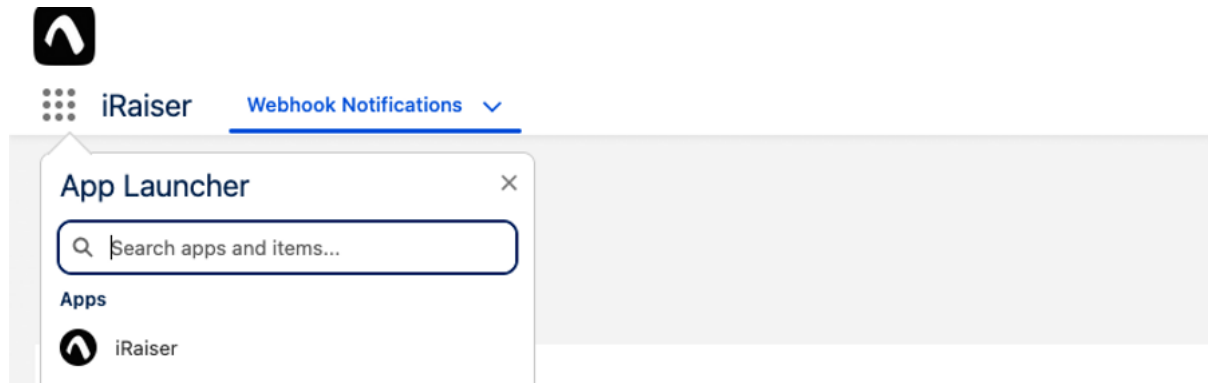


Assign the iRaiser Webhook permission set to the guest user.

You should now be ready to receive webhook notifications. To modify the mapping of the webhook notifications, see **Mapping Webhook Notifications**.

Step 5: Assign Admin Permissions

To allow users to manage incoming notifications and debug them, access to the iRaiser app is needed. For this, assign the **iRaiser Admin** permission set. Users with this permission set have access to the app and to the custom object that captures the notifications, prior to handling them.



The iRaiser Admin permission set.

Monitoring and Management

Incoming notifications can be monitored in the **Webhook Notifications** tab. Each incoming notification is stored here first. The object has a create trigger, which starts the enrichment and the execution of the handler flow as configured in the iRaiser settings.



Webhook Notifications list view.

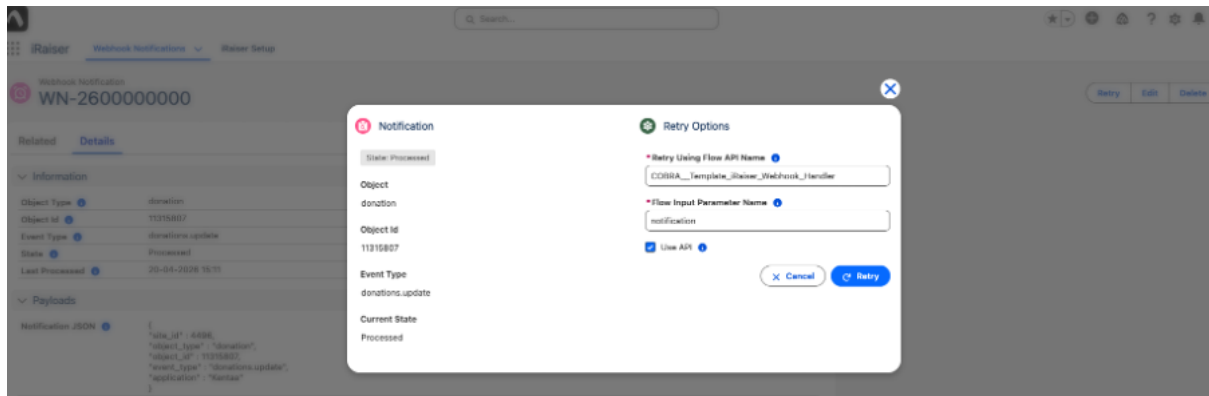
The state determines whether a notification has been offered to the designated handler flow successfully. It does not hold the result of the flow in question. In other words, the state can be **Processed** while the flow itself may have failed. If the state is **Error**, the **Last Run Messages** field shows the error that occurred.

Retry

The page for Webhook Notifications shows a **Retry** button. When clicked, the pop-up allows the admin to retry processing of the notification. By default the flow and parameter name are taken from the configuration, but they may be overridden to use a different flow.

The **Use API** field is only available when the **Full Payload** field has been populated by a previous enrichment. By default this option is checked, which means the processor fetches the latest state of the payload from the iRaiser API. If unchecked, the processor uses the **Full Payload** value from the previous enrichment. This allows the admin to re-process notifications based on the state of the last enrichment.

Caution Re-processing from a stored payload may process old data that has changed in the meantime on the iRaiser platform, overwriting newer data in Salesforce.



The Retry dialog on a webhook notification.

Mapping Webhook Notifications

The package supports notifications for the following inbound objects:

1. Donation
2. Company
3. Segment
4. Project
5. Team
6. Action
7. Newsletter Subscription

The mapping of incoming messages is handled by the flow configured in **Notification Handler Flow** of the iRaiser Settings. By default this is the template flow that comes with the package, **COBRA__Template_iRaiser_Webhook_Handler**.

This flow can be used as a starting point to create a custom handler flow by opening the template and using **Save As New Flow**.

Please note: if you are using the **Salesforce Nonprofit Cloud**, you can find a premade NPC Template Flow in the AgentExchange listing, under the Resources Tab!

**iRaiser for Salesforce** ❤️
Integrate your iRaiser Community Fundraising Platform data with Salesforce!

This listing is private and can't be found by searching or browsing AgentExchange.

Free
Never requires payment
[Get It Now](#)
[Try It](#)

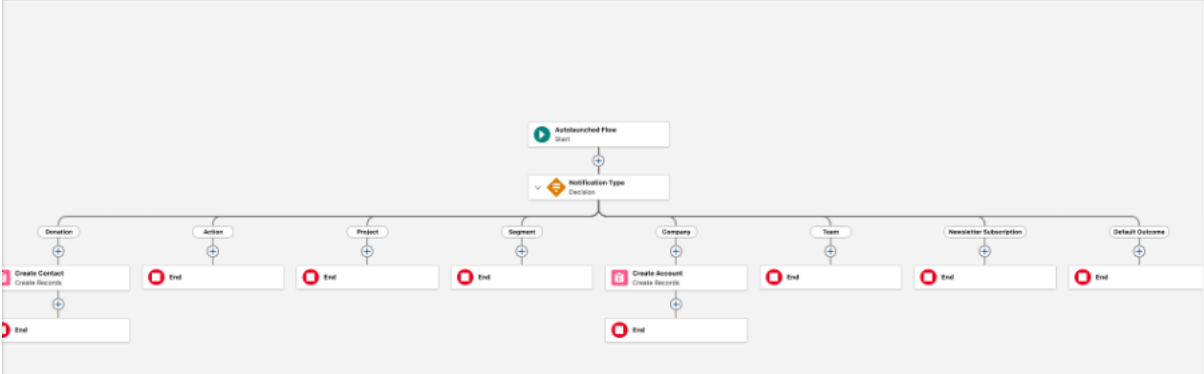
[Overview](#) [Specs](#) [Security](#) [Reviews](#) **[Resources](#)** [Support](#)

**Guides**
[Admin Manual](#)

**Data Sheets**
[One Pager](#)
[FAQ](#)
[NPC Template Flow](#)

Flow Builder | Template Raiser Webhook Handler - V1 | Run as System

Last saved on 20-4-2026, 11:41 | Active | Run | Debug | Save As New Flow | Save | Deactivate



The packaged template handler flow.

The flow needs to have a variable available for input, with the name configured in **Parameter Name** in the iRaiser Settings. The default name is **notification**; the type should be **COBRA__ExtendedNotification**.

iRaiser Fundraising for Salesforce — Admin Manual

8

×

Edit Variable

• API Name ⓘ

notification

Description

Input for the flow - single notification

• Data Type ⓘ

Apex-Defined ▼

☐ Allow multiple values (collection) ⓘ

• Apex Class ⓘ

COBRA__ExtendedNotification

Availability Outside the Flow

☒ Available for input

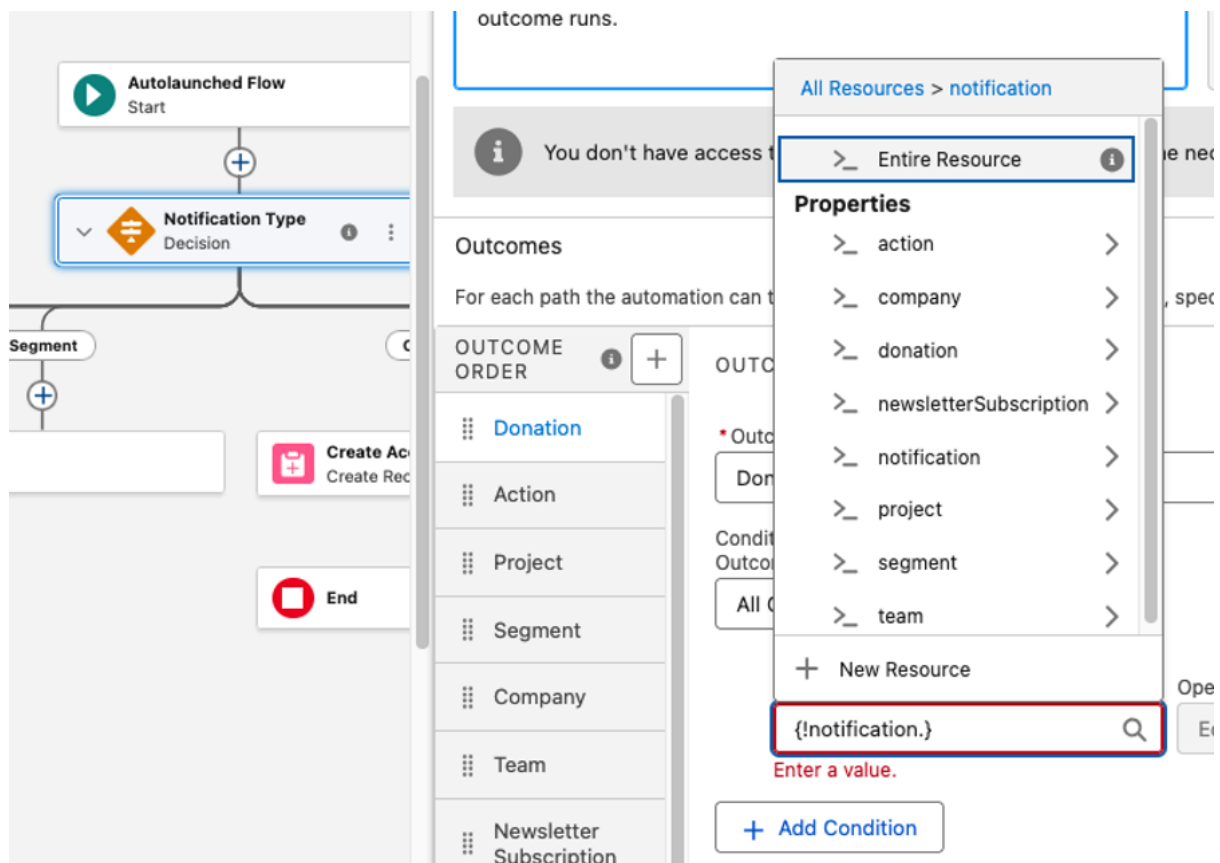
☐ Available for output

Cancel

Done

Input variable of the handler flow.

Each **COBRA__ExtendedNotification** contains the notification information and the payload of the appropriate object. Use these in the flow to map the incoming notification to the correct objects and fields, and to apply the right business logic.



Notification and payload available inside the flow.

When modifying the flow and saving it as a new flow, make sure to update the flow name in the iRaiser Settings. This ensures that the next webhook notification is handled by your new flow.

Cleanup

To automatically delete Webhook Notification records that have been processed, schedule the Apex class **WebhookNotificationCleanupScheduler** under **Setup → Environments → Jobs → Scheduled Jobs**.

The example in the screenshot below starts the cleanup process every day at 3 a.m.

The screenshot shows the Salesforce Setup interface for configuring an Apex Class. The left sidebar contains a search bar with 'sched' and a list of categories: Feature Settings, Sales, Products, Environments, and Jobs. The main content area is titled 'Apex Classes' and shows the configuration for a job named 'Clean up iRaiser notifications'. The job uses the 'WebhookNotificationCleanup' class and is scheduled using the 'Schedule Builder'. The frequency is set to 'Weekly' with a recurrence every week on all days of the week (Sunday through Saturday). The start date is '21-04-2026' and the end date is '23-06-2030'. The preferred start time is '03:00'. A note at the bottom states: 'Exact start time will depend on job queue activity.'

Scheduling the cleanup job.

The cleanup process deletes all Webhook Notification records where the **Last_Processed__c** date is older than the configured number of **Retention Days**.

Only records in state **Processed** are deleted. Records in state **New** or **Error** remain and need to be updated to state **Processed** to be deleted as well. The same applies to state **Retain**, which is intended to keep certain records for auditing and other purposes.

Need help?

Contact Cobra for support through support@cobracrm.nl