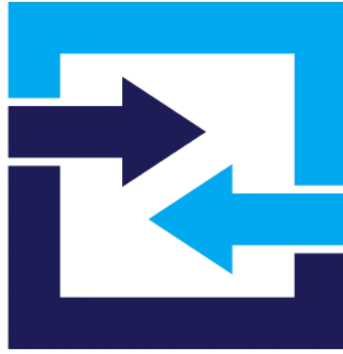




CYFUNO LABS

Microsoft Teams Integration Documentation

User Guide - AppExchange Product

Last Modified: January 17, 2025

For the latest documentation, it's always best to check out the live version at

<https://www.cyfunolabs.com/vikundi/docs/>

Table of Contents

Table of Contents.....	2
Getting Started.....	4
Package Installation.....	4
Create Salesforce Site.....	7
Register Site's Domain.....	7
Create the Site.....	8
Edit Public Access Settings.....	9
Create Bot in Teams.....	11
Create Bot in Microsoft Teams.....	11
Register the Bot in Salesforce.....	13
Adding Bot API Permissions in Teams.....	14
Create Teams App.....	16
Add Bot to a Team.....	18
Testing Notifications.....	19
Test a Simple Notification.....	19
Test a Structured Notification.....	20
Test an Advanced Notification.....	21
Assigning Permissions.....	23
Sending Notifications.....	23
Send a Simple Notification.....	23
Permissions Required.....	24
Send a Structured Notification.....	26
Permissions Required.....	27
Send an Advanced Notification.....	29
Create your Design.....	29
Create your Flow.....	31
Merge Variables.....	35
Tokens.....	35
Mention Users and Tags.....	35
Examples.....	36
Formatting with Markdown.....	37
Available Formatting.....	37
Examples.....	37
Working with Channels.....	38
Create Channel Action.....	38
Permissions Required.....	38
Add User to Channel Action.....	41
Permissions Required.....	41
Archive Channel Action.....	44
Permissions Required.....	44

Delete Channel Action.....	46
Permissions Required.....	46
Page Components.....	48
Presence Display Component.....	48
Permissions Required.....	48
Add Component to Page.....	48
Presence Caching.....	51
Managing AppExchange Subscription.....	51
Permissions Required.....	51
Billing Details.....	51
Invoices / Receipts.....	51
Updating the Package.....	52
Uninstalling the App.....	54

Getting Started

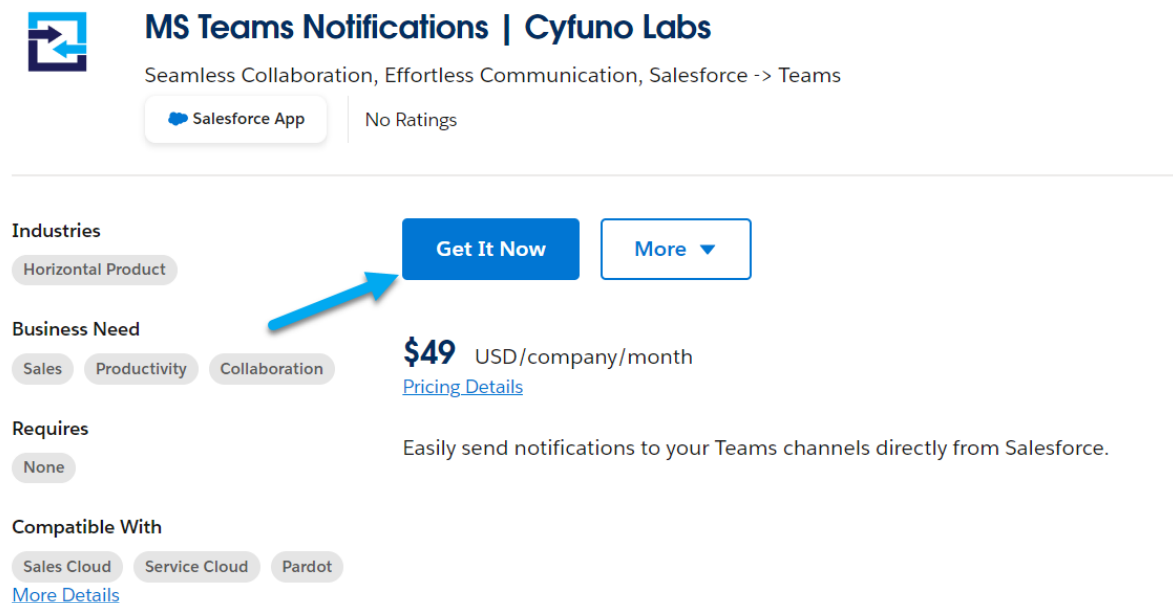
Embark on a straightforward journey in configuring the Salesforce to Teams Integration. Whether you're a Salesforce or Teams administrator, this step-by-step guide ensures a smooth and hassle-free installation process, allowing you to quickly unlock the full potential of cross-platform communication.

We've included a version of these instructions directly in the Setup Assistant, as we know first-hand how challenging it can be to find (and read) the docs. On this page we've included additional Screenshots and guidance should you get stuck. Also, the Setup Assistant will provide contextual links whenever possible, taking you right to the place you need to be.

Package Installation

Installation begins with the AppExchange. From our Listing, click the Get It Now button.

You may need to Login to the AppExchange.



The screenshot shows the AppExchange listing for the 'MS Teams Notifications | Cyfuno Labs' package. The header includes the package icon, name, and description: 'Seamless Collaboration, Effortless Communication, Salesforce -> Teams'. Below this, it indicates it is a 'Salesforce App' with 'No Ratings'. The main section features a 'Get It Now' button, which is highlighted by a blue arrow, and a 'More' button with a dropdown arrow. To the left of the 'Get It Now' button, there are filters for 'Industries' (Horizontal Product), 'Business Need' (Sales, Productivity, Collaboration), 'Requires' (None), and 'Compatible With' (Sales Cloud, Service Cloud, Pardot). The pricing is listed as '\$49 USD/company/month' with a link to 'Pricing Details'. A 'More Details' link is also present at the bottom left.

Verify that the Salesforce Username is correct, then choose if you want to Install into Production or in a Sandbox.

Where do you want to install this package?

Install in a Production Environment

Install this package in the org where you or your users work, including Developer Edition orgs.

* Connected Salesforce Accounts ⓘ

support@cyfunolabs.com

Don't see your account? [More Info](#)

 **Install in Production**

Install in a Sandbox

Test this package in a copy of a production org.

Install in Sandbox

Cancel

Next, provide your contact and billing details.

AppExchange Checkout

Let's get started

To install this product, we need your contact and billing details. If the provider offers a free trial, you won't be charged until the trial ends.



* First Name ⓘ * Last Name

* Email

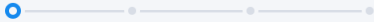
Billing Address

* Country

* Street

* City

* State * Zip/Postal

 **Next**

Choose your payment plan.

(in our screenshot we only have the Monthly plan, but there are others)

AppExchange Checkout

Choose your payment plan

Choose the payment plan that works best for you. Then tell us the number of user licenses you'd like.

☒ \$49 USD per month

Coupon Code

Subtotal \$49.00
 Discount -\$0
Payment (Monthly) \$49.00



Back
Next

Provide your payment information. This can be updated later if the card changes or becomes expired.

AppExchange Checkout

Add a payment method

You can pay with a credit card, or transfer funds from a bank account.

Credit Card

* Card Number
 * CVC

* Expires



Back
Next

On the Confirm Installation Details screen, make sure everything is correct (scroll down to double check everything). You'll also be asked a second time to "really" confirm.

Confirm Installation Details

support@cyfunolabs.com

Here are the details we'll share from your profile

* First Name	Support	* Company	Cyfuno Labs
* Last Name	Team	* Country	United States
Job Title		* State/Province	Delaware
* Email	support@cyfunolabs.com		
Phone			

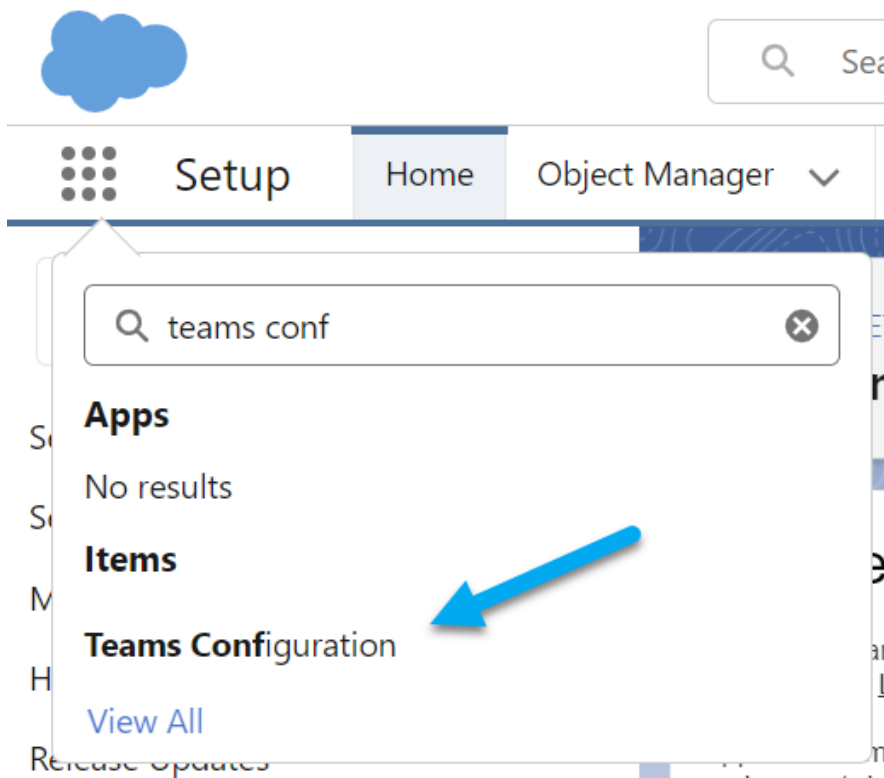
[Edit Profile](#)

☒ I have read and agree to the terms and conditions.

☒ Allow the provider to contact me by email, phone, or SMS about other products or services I might like

Cancel
Confirm and Install

When the installation is complete, launch the Teams Configuration from the App Launcher.



Note: If you are delegating the actual app configuration to another user, they will need to either be:

- A user with the Salesforce Administrator profile
- A user with the Packaged Permission Set: **Teams Notification Admin**

Create Salesforce Site

A Salesforce Site is needed to allow Microsoft Teams to send Salesforce some key events, such as our Bot being registered in a particular Team.

Instructions and values you can use are provided in the Assistant in the middle column.

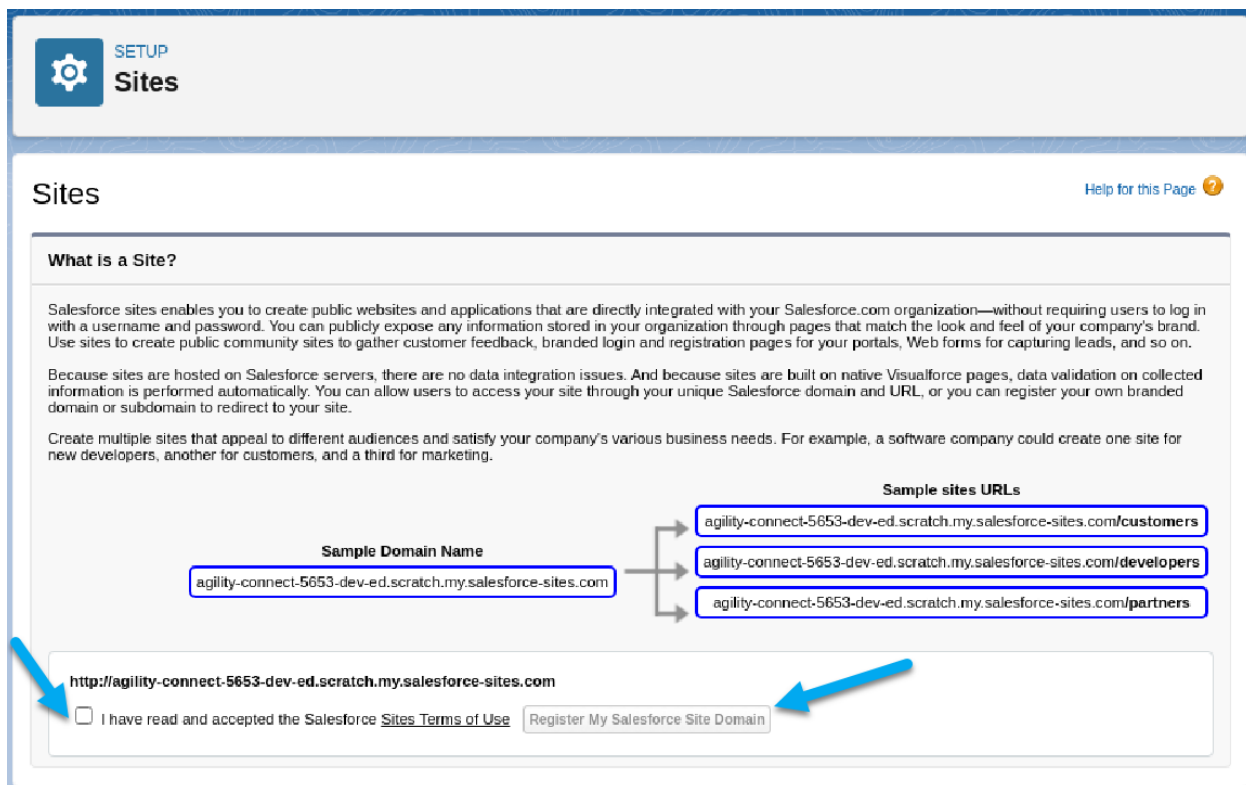
As the Site is configured, you will notice bullet points appear on the left showing you what pieces are not yet complete. This section refreshes automatically every few seconds to give you an up-to-date view of the installation.

Register Site's Domain

This only needs to be done once per Salesforce Org, and if you have already registered your site's domain, you can move on to Create the Site.

Permissions Needed: Customize Application. Salesforce Docs

1. From Salesforce Setup, navigate to **Sites** using either Quick Find or User Interface > Sites and Domain > Sites.
2. Read and Accept the Salesforce Sites Terms of Use.
3. Click on the button to "Register My Salesforce Site Domain".
4. Confirm that the domain you are registering is the one you want. This cannot be changed after-the-fact, so be double sure!
5. Wait for Salesforce to do things in the Background.



Create the Site

This only needs to be done once per Salesforce Org, and if you have already created a Site you can move on to Editing Public Access Settings.

Permissions Needed: Customize Application, Manage Users. Salesforce Docs

1. From Salesforce Setup, navigate to **Sites** using either Quick Find or User Interface > Sites and Domain > Sites
2. Create a new Site, filling in a few specific values:
 1. **Site Label:** Teams Endpoint
 2. **Site Name:** Teams_Endpoint
 3. **Site Description** (optional): Enables the Cyfuno Labs MS Teams Integration to work.
 4. **Active:** Checked
 5. **Active Site Home Page:** TeamsAdminInstructions

3. Click Save. It's likely the page may appear to do nothing. Just give it some time and Salesforce will let you know when it was successful.

Site Edit [Help for this Page](#)

New Site

Site Label: [i](#)

Site Name: [i](#)

Site Description:

Site Contact: [i](#)

Default Record Owner: [i](#)

Default Web Address: [i](#)

Active: ☒ [i](#)

Active Site Home Page: [i](#)

Inactive Site Home Page: [i](#) [\[Preview\]](#)

Site Template: [i](#)

Site Robots.txt:

Edit Public Access Settings

Next, edit the Public Access Settings of the Site.

- From the Site that was just created, click the Public Access Settings button.

Site Details [Help for this Page](#)

Teams Endpoint

[Back to List: Sites](#)

Site Detail

Site Label: **Teams Endpoint** Site Name: **Teams_Endpoint**

Looking at the Site's Guest User Profile, we want to enable Apex Class access to 2 Classes.

- Click on Enabled Apex Class Access, then click Edit.

Profile [Help for this Page](#)

Teams Endpoint Profile

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

[Login IP Ranges \[0\]](#) | [Enabled Apex Class Access \[0\]](#) | [Enabled Visualforce Page Access \[12\]](#) | [Enabled External Data Source Access \[0\]](#) | [Enabled Named Credential Access \[0\]](#) | [Enabled External Credential Principal Access \[0\]](#) | [Enabled Custom Metadata Type Access \[0\]](#) | [Enabled Custom Setting Definitions Access \[0\]](#) | [Enabled Flow Access \[0\]](#) | [Enabled Custom Permissions \[0\]](#)

- Add cfl_teams.TeamsAdminController.
- Add cfl_teams.TeamsBotController.
- Click Save.

Again looking at the Site's Guest User Profile, we want to give access to 2 site Objects.

- Click on Edit to Edit the Profile.

- For the Custom Object "Teams Bots", click Create.
- For the Platform Event "Teams Bot Inbound Notification", click Create.
- Scroll to the bottom of the page, click Save.

Object	Basic Access	Data Administration
	Read Create Edit Delete	View All Modify All
Teams Bots	☒ ☒	☐ ☐
Teams Channels	☐ ☐	☐ ☐
Teams Bot Inbound Notifications	☒ ☒	☐ ☐
Teams Bot Outbound Notifications	☐ ☐	☐ ☐

Create Bot in Teams

With an active Site available in Salesforce, a Microsoft Teams Administrator can follow a few steps to get things configured.

The Setup Assistant will provide a link to the Teams Instructions that you can give to a Microsoft Teams Administrator. They will be able to use the Instructions page (without a Salesforce User License) to securely provide Salesforce with the information needed.

As a Microsoft Teams Administrator, you will likely start by seeing the Salesforce & Teams Bot Registration page that was sent to you.

- Get Started by clicking on the Bots page link, which will take you to your Bots Management page in the Developer Portal in Microsoft Teams.

Salesforce & Teams Bot Registration

As part of the installation of the Salesforce & Teams integration, a Teams Bot and App will need to be created.

What information will flow between the systems?

- Teams will send event notifications to Salesforce when the Bot is added to a new channel. No user information is transmitted.
- Notifications from Salesforce will flow into desired Teams Channels where the Bot has been added.

Information Collection

Bot Name

Name of the Bot as you want it to appear within Teams.

Bot Id

ID of the Bot from the Tools section.

Bot Secret

Value of the Secret that was created for this bot.

[Register Bot](#)

Instructions

First, we need to create a Bot.

1. Go to the Teams Developer Portal, [Bots page](#). It can sometimes take a few moments for the list to populate.
2. Create a New Bot by clicking the **New Bot** button at the top
3. Give the Bot a name, this will be displayed in the Channel when a message is sent. Provide the name to the **Name** field on this page
4. Click **Add**
5. Grab the Bot ID from the URL. It is a 36 character, Globally Unique ID. Paste it into the **Bot Id** field on this page.
6. Create a Client secret. Paste the secret into the **Secret** field on this page. **Beware:** sometimes Team's copy button doesn't work, double check what you are pasting.
7. Click OK to dismiss the newly created Secret
8. Click the **Register Bot** button on this page.

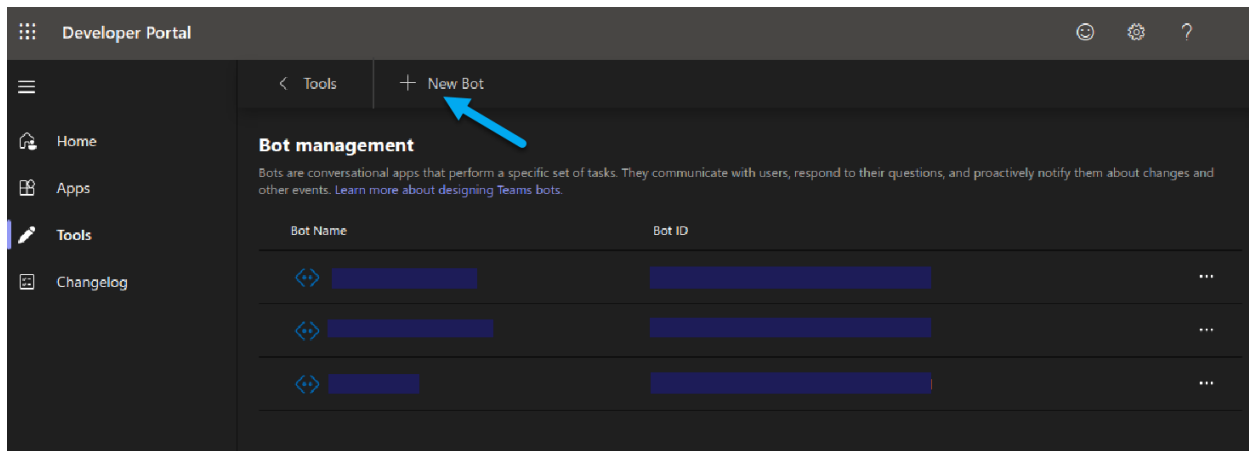
We now have a shell of a Bot, and should have collected all 3 values required by the integration. Next we will save this information and finish the Bot configuration

Create Bot in Microsoft Teams

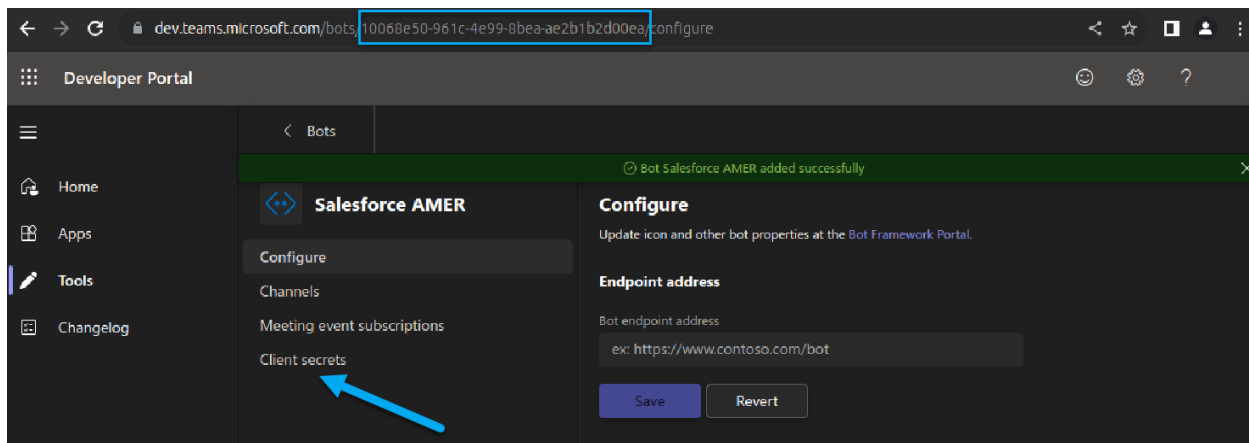
Create a New Bot

- Click the New Bot button at the top.

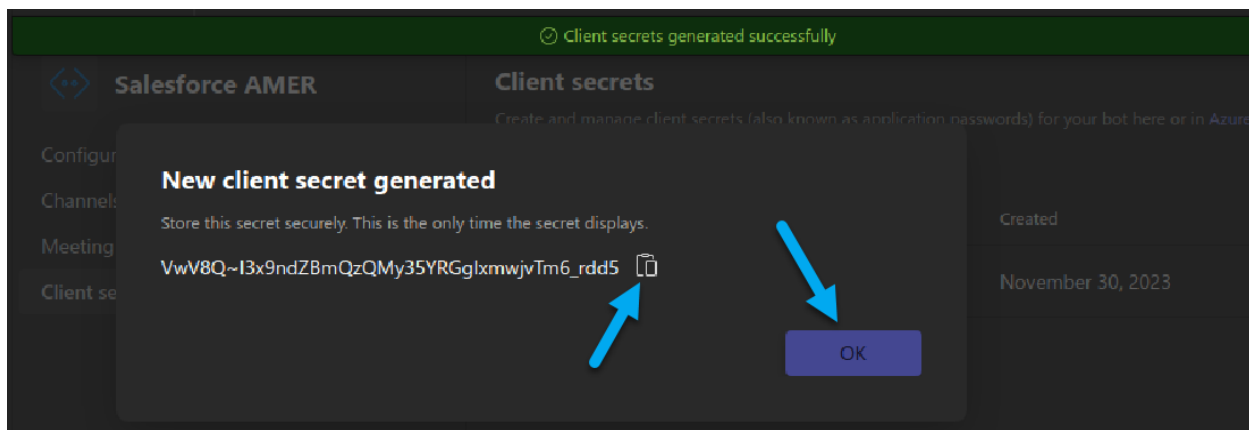
- Give it a Name. This name will appear whenever the Bot sends a Message.
- Click Add.
- Back at the Instructions page, provide the Name you just used.



- Next we will need the Bot Id.
- In the Browser Address Bar, copy the Id to your clipboard. It is a 36-character UUID.
- Back at the Instructions page, paste this Id into the Bot Id field.
- Click on Client Secrets.



- "Add a client secret for your bot".
- Copy the secret to your clipboard, click Ok.
- Back at the Instructions page, paste this secret into the Bot Secret field.
- (optional) click the eye icon to verify that you are pasting the secret.



Register the Bot in Salesforce

- With all information collected in the Instructions Page, click the Register Bot button.
- Next, we need to a minor adjustment to the automatically-generated Bot Registration. Click on the Bot Application Page link.

Bot Id

7d3435fd-3874-481e-993f-247d592f1ae

ID of the Bot from the Tools section.

Bot Secret

Value of the Secret that was created for this bot.

3. Give the Bot a name, this will be displayed in the Channel when a message is sent. Provide the name to the **Name** field on this page
4. Click **Add**
5. Grab the Bot ID from the URL. It is a 36 character, Globally Unique ID. Paste it into the **Bot Id** field on this page.
6. Create a Client secret. Paste the secret into the **Secret** field on this page. **Beware:** sometimes Team's copy button doesn't work, double check what you are pasting.
7. Click OK to dismiss the newly created Secret
8. Click the **Register Bot** button on this page.

We now have a shell of a Bot, and should have collected all 3 values required by the integration. Next we will save this information in Salesforce and get ready for next steps.

Authentication Test

Test Auth

Instructions

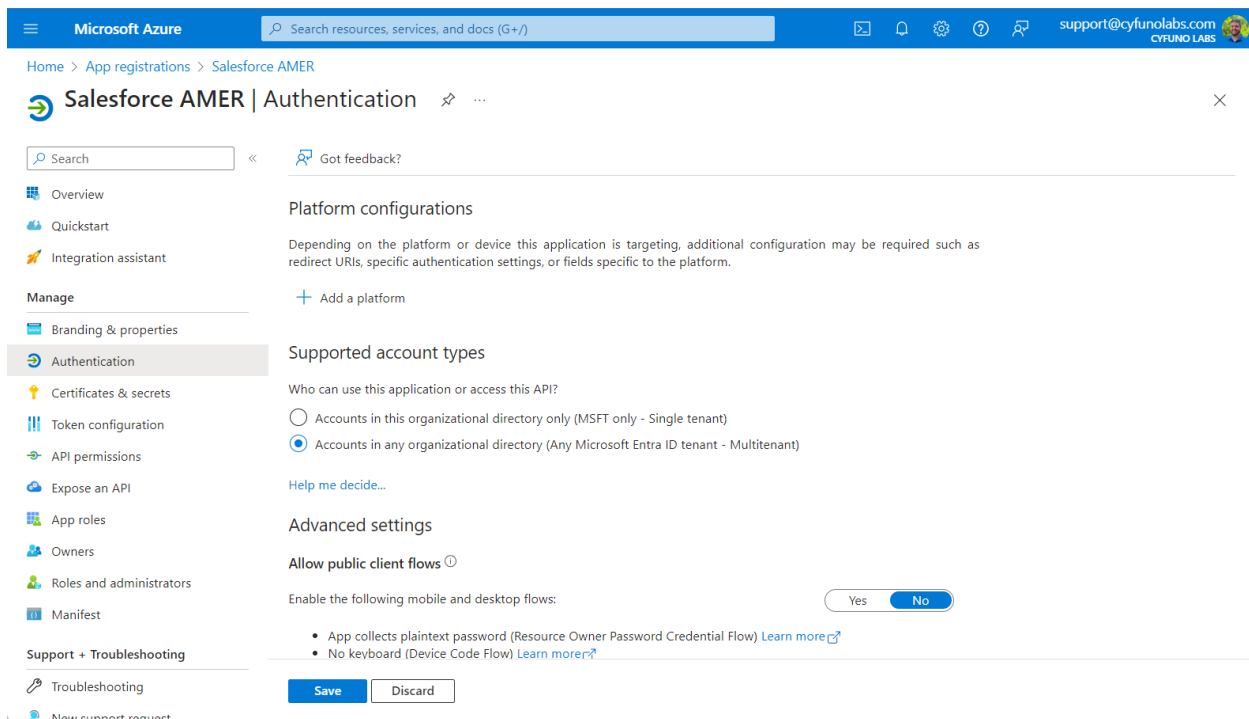
Next, we need to adjust a Bot Setting.

1. Go to the Azure Portal, [Bot Application Page](#).
2. Under Supported account types, change to Multitenant. While it won't actually be treated as such, it is required for certain API calls to be allowed.
3. Click **Save** at the bottom of the screen.
4. It may take a few moments for the change to propagate within Azure.
5. Click the **Test Auth** button on this page to verify all is working.

We now have communications working between the 2 platforms. Next we need to configure Teams to allow the Bot to be added to Channels

- For the Bot Authentication, select the Multitenant option.
- Save this change, close the browser tab.

We make this change so that the Salesforce -> Teams API calls can work.



Adding Bot API Permissions in Teams

Next, we need to add some API permissions.

On the Azure Portals Registration page for the app, go to **API Permissions**.

Click **Add a permission**, select **Microsoft Graph**, then select **Application permissions**.

- Add Channel.ReadBasic.All (this allows us to retrieve the Active Channels on the Team the bot was added to).

(optional) Additional features can be enabled by granting API Permissions. Select the permissions for the features you would like to allow

1. Channel.Create & Team.ReadBasic.All -> Allows the bot to create a Channel for the Teams it has been added to.
2. Channel.Delete.All -> Allows Salesforce to delete a Channel.
3. ChannelMember.ReadWrite.All -> Allows the bot to add/remove a Teams User to a Shared/Private Channel.
4. ChannelSettings.ReadWrite.All -> Allows the bot to Archive a Channel, within the Teams it has been added to.
5. Presence.Read.All -> Allows the retrieval of Teams User's statuses / presence indicators.
6. TeamworkTag.Read.All -> Allows the @mentioning of a Team's Tag by its Display Name.

7. User.ReadBasic.All -> Allows the @mentioning of a User by their Display Name or Email address.

Once permissions have been added, Grant admin consent for all API permissions.

Microsoft Azure

Home > App registrations > May19a

Salesforce AMER | API permissions

Search resources, services, and docs (G+J)

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

Support + Troubleshooting

Troubleshooting

New support request

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for MSFT

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Microsoft Graph (5)				
Channel.Create	Application	Create channels	Yes	Granted for MSFT
Channel.ReadBasic.All	Application	Read the names and descriptions of all channels	Yes	Granted for MSFT
ChannelSettings.ReadWrite.All	Application	Read and write the names, descriptions, and settings of all...	Yes	Granted for MSFT
TeamworkTag.Read.All	Application	Read tags in Teams	Yes	Granted for MSFT
User.ReadBasic.All	Application	Read all users' basic profiles	Yes	Granted for MSFT

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

- Back at our Instructions page, click the Test Auth button to test Salesforce -> Teams authentication.
- With a successful test, scroll down a bit to focus on the next set of instructions.
- Copy the Endpoint Address.

Download Manifest

Finish wiring up the Bot, and create the App.

Endpoint Address

`https://cf.cyfunolabs.com/notify?t=https://app-speed-5423-dev`



Download manifest.zip

Instructions

Finishing up the Teams Bot configuration.

1. Copy the endpoint address from this page.
2. Back in the Teams Developer Portal, navigate to the **Configure** page for the Bot just created
3. Paste the Endpoint Address, click **Save**

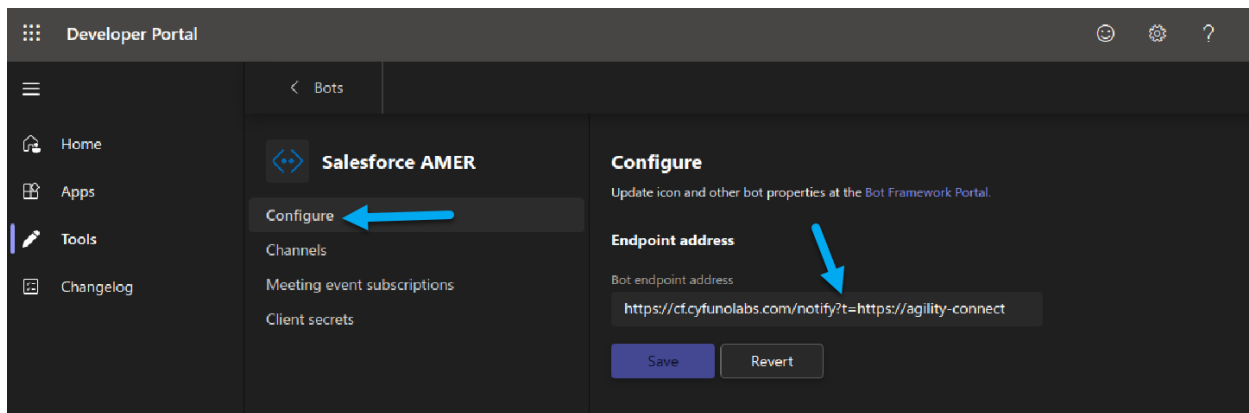
Now that the Bot has been created, we need a Teams App to allow the messages to appear in Channels

1. Download the **manifest.zip** from this page. It is specific to this the Teams Bot you just created.
2. Navigate to your [Microsoft Teams "Apps" page](#)
3. Click on **Manage your apps** at the bottom of the left-hand navigation.
4. Click on **Upload an app**
5. Select **Upload an app to your org's app catalog**
6. Choose the manifest.zip file that you just downloaded.
7. It can take 1 hour to a couple business days for Microsoft to do its thing in the background.
8. You can look at the **Pending Requests** from the **Manage your apps** page to see what you've created worked and is in the queue.

At this point you can add the Bot to a Team, or feel free to add it later, using the instructions below.

1. In Teams, go to [Teams Apps page](#).
2. Click **Built for your org**.
3. Click the Name of the Bot in the Bot's card.
4. Click **Add to a team**

- Back in the Teams Developer Portal, click on the Configure side-tab.
- Paste the Bot endpoint address, click Save.
- Do not leave / refresh this page, as this information is only available once!



Create Teams App

Adding an App in Microsoft Teams is what allows the Bot to be added to a Team / Channel.

- From the instructions page, Download the manifest.zip file.
- Navigate to the Teams Apps page from the main Teams UI's left-hand menu.

Download Manifest

Finish wiring up the Bot, and create the App.

Endpoint Address

<https://cf.cyfunolabs.com/notify?t=https://app-speed-5423-dev>

 Download manifest.zip

Instructions

Finishing up the Teams Bot configuration.

1. Copy the endpoint address from this page.
2. Back in the Teams Developer Portal, navigate to the **Configure** page for the Bot just created
3. Paste the Endpoint Address, click **Save**

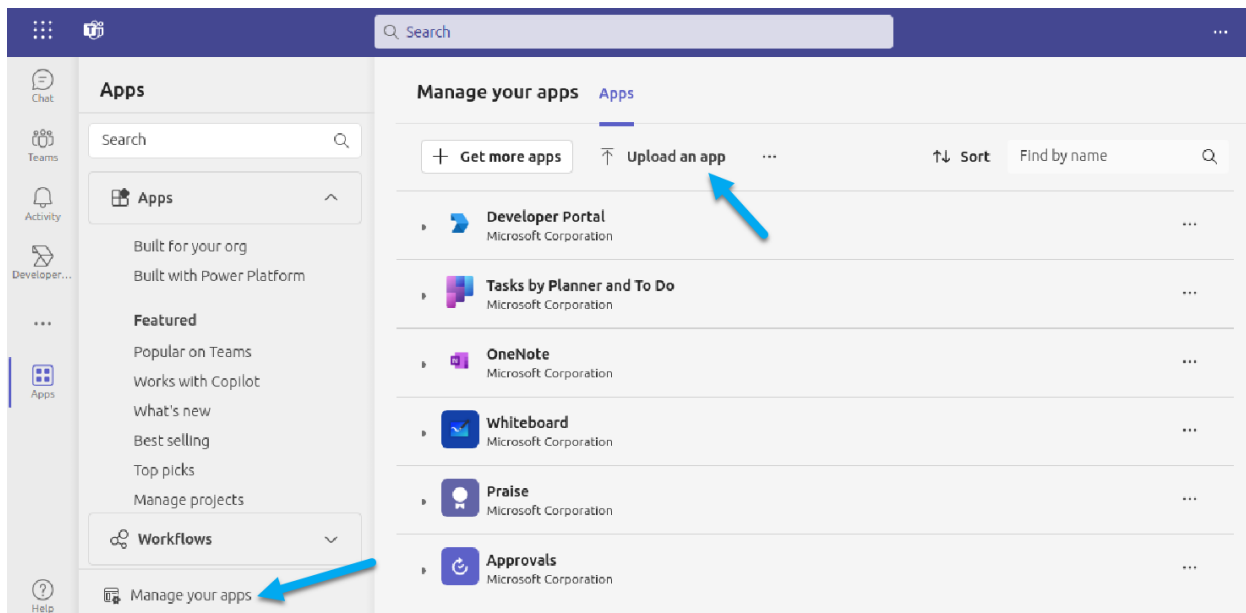
Now that the Bot has been created, we need a Teams App to allow the messages to appear in Channels

1. Download the **manifest.zip** from this page. It is specific to this the Teams Bot you just created.
2. Navigate to your [Microsoft Teams "Apps" page](#)
3. Click on **Manage your apps** at the bottom of the left-hand navigation.
4. Click on **Upload an app**
5. Select **Upload an app to your org's app catalog**
6. Choose the manifest.zip file that you just downloaded.
7. It can take 1 hour to a couple business days for Microsoft to do its thing in the background.
8. You can look at the **Pending Requests** from the **Manage your apps** page to see what you've created worked and is in the queue.

At this point you can add the Bot to a Team, or feel free to add it later, using the instructions below.

1. In Teams, go to [Teams Apps page](#).
2. Click **Built for your org**.
3. Click the Name of the Bot in the Bot's card.
4. Click **Add to a team**

- Click Manage your apps
- Click Upload an app.
- In the dialog that appears, choose "Upload an app to your org's app catalog".
- Select the manifest.zip file you just downloaded a few steps earlier.
- It can take 1 hour to a couple business days for Microsoft to do its thing in the background.
- You can look at the Pending Requests from the Manage your apps page to see what you've created worked and is in the queue.



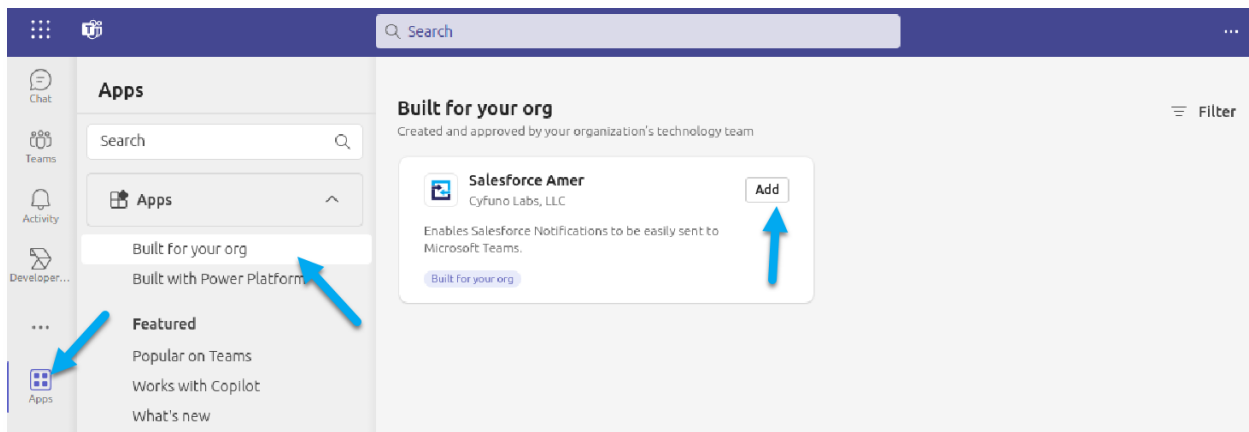
Add Bot to a Team

With a Bot created in your Teams Instance, it can now be added to a Team that has Channels you want to send notifications into.

Depending on your Company, this step might be performed by either the Salesforce Admin (if they have enough Teams permissions), or it may need to be performed by a Teams Administrator.

- In MS Teams, Click on Apps in the left-side menu.
- Next, choose "Built for your org" to make finding the Bot easier.
- Click the Add button in the Bot that you created.

Note: if you get an error that the App cannot be found, this is due to not waiting long enough after initially adding it.



- Click the "Add to a team" button
- Choose a Team by either typing in the Team Name, or a Channel within that Team.
- Click the "Set up a bot" button to confirm.
- Repeat as necessary to add to other Teams



Salesforce Amer

Cyfuno Labs, LLC

Add to a team



Overview Permissions Discover more apps

Enables Salesforce Notifications to be easily sent to Microsoft Teams.

Once added to a Team, its Channels can be selected as a target for notifications within Salesforce. This is not a conversational bot, it is push only.

App features

Bots

Chat with the app to ask questions and find info

Created by: [Cyfuno Labs, LLC](#)

Version 1.0.1

By using Salesforce Amer, you agree to the [privacy policy](#), [terms of use](#), and [permissions](#).

Testing Notifications

Now it's the moment of Magic, we are ready for our first test.

There are 3 different kinds of Message styles that can be sent from the Testing panel on the right-hand side of the assistant.

These tests are designed to be as quick and easy as possible, with some default messages and values. Just pick the Channel you are testing with and Send away.

Test a Simple Notification

From the Teams Configuration page in Salesforce, use the Notifications Channel Testing section on the right hand side.

- Select a Teams Channel to send the notification into by typing its name.
- Adjust the Test Message if desired.
- Click Send Test Notification.
- Wait for the Confirmation Message under the button.

For additional customization, visit our [Send a Simple Notification](#) page.

Microsoft Teams Notifications Channel Testing

✓ Send a Simple Notification

Click the button below to send a test notification to the channel.

*Channel

 Notifications 

*Test Message

This is a simple test message from the Microsoft Teams Notifications Setup Assistant in Salesforce.

Send Test Notification

Test a Structured Notification

From the Teams Configuration page in Salesforce, use the Notifications Channel Testing section on the right hand side.

- Open the "Send a Structured Notification" area.
- Select a Teams Channel to send the notification into by typing its name.
- Adjust the Test Title and Message if desired.
- Change / Remove Fact Titles and Values in Pairs. (highlighted by light-blue box)
- Change / Remove Action Labels and URLs in Pairs. (dark-blue box)
- Click Send Test Notification.
- Wait for the Confirmation Message under the button.

For additional customization, visit our [Send a Structured Notification](#) page.

Microsoft Teams Notifications Channel Testing

> Send a Simple Notification

Send a Structured Notification

Click the button below to send a test notification to the channel.

* Channel

Notifications

* Title

Test Message

* Message

This is a **structured test message** from the **Microsoft Teams Notifications Setup Assistant** in Salesforce.

Fact Title 1

Org Id

Fact Value 1

00D8H000

Fact Title 4

Support D

Fact Value 4

[Cyfuno S

Fact Title 2

Assistant

Fact Value 2

User User

Fact Title 5

Billing Det

Fact Value 5

[AppExch

Fact Title 3

Message

Fact Value 3

[Using Ma

Fact Title 6

Fact Value 6

Action Label 1

Setup Assistant

Action Label 2

Salesforce Site

Action Label 3

Action URL 1

https://agility-cor

Action URL 2

Action URL 3

Send Test Notification

Test an Advanced Notification

From the Teams Configuration page in Salesforce, use the Notifications Channel Testing section on the right hand side.

- Open the "Send a Custom Adaptive Card" area.
- Select a Teams Channel to send the notification into by typing its name.
- Use the existing Adaptive Card Payload, or create your own by going to Microsoft's Adaptive Card Designer and copying the card payload.
- Click Send Test Notification.
- Wait for the Confirmation Message under the button.

For additional customization, visit our [Send an Advanced Notification](#) page.

Pg. 21

Microsoft Teams Notifications Channel Testing

> Send a Simple Notification

> Send a Structured Notification

> Send a Custom Adaptive Card (Advanced)

Click the button below to send a test notification to the channel.

* Channel

Notifications

✕

* Adaptive Card Payload

```
{
  "type": "AdaptiveCard",
  "body": [
    {
      "type": "TextBlock",
      "size": "Medium",
      "weight": "Bolder",
      "text": "Publish Adaptive Card Schema"
    },
    {
      "type": "ColumnSet",
      "columns": [
        {
          "type": "Image",
          "style": "Person",
          "url": "https://pbs.twimg.com/profile_images/3647943215/d7f12830b3c17a5a9e4afcc370e3a37e_400x400.jpg"
        }
      ]
    }
  ]
}
```

Send Test Notification

Assigning Permissions

There are 3 Permission Sets that are included with this app. Assign them to users to allow them to perform certain actions.

Permission Set Name	Description
Teams Notification Admin	Allows users to configure the app through the Setup Assistant.
Teams Notification Channel CRED	Allows users to make changes to Teams Channels. Create, Add Members, Archive, Delete.
Teams Notification Designer	Allows users to build Flows with the Actions provided by this app.

Everything is set up! Next it's time to begin building your notifications.

Sending Notifications

Channel - Simple Notification

Simple notifications are ideal when you are needing to send a line or two of text.

Great examples include:

- Process Status Notifications
- Error Notifications
- Simple Alerts

If this is your first Notification with our integration, we highly recommend starting here before progressing to other styles.

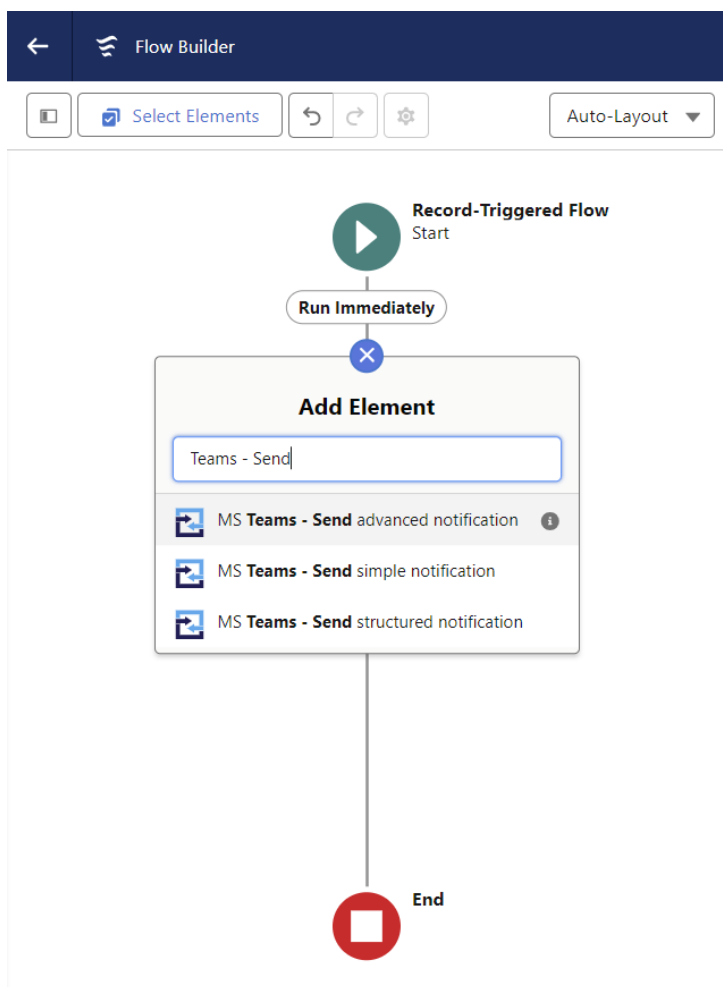
Permissions Required

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

Note: Notifications can only be sent to Standard Channels.

From your Flow, we will start by adding a new Element.

- Search for Teams, select MS Teams - Send Simple notification



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Notification.
- Search for the Teams Channel you want this notification to appear in.
- Type in the Message you wish to have displayed.
- Click **Done** to save this Action to the Flow designer.
- Don't forget to Save & Activate the Flow when you are ready!

The message can be in **Plain Text**, or it can be composed using **Markdown**.

The screenshot shows a 'New Apex Action' dialog box. At the top, there is a 'Record-Triggered Flow Start' icon and a 'Run Immediately' button. The dialog has a title bar with a close button. The main content area includes:

- * Label:** A text input field containing 'Send Notification'.
- * API Name:** A text input field containing 'Send_Notification'.
- Description:** A text area containing the text: 'This example uses a Text Template for the Message. It allows us to easily use Formulas, Functions and Variables within the message.'
- Teams Simple Notification Information:**
 - * Channel:** A dropdown menu showing 'Notifications' with a close button.
 - Message for Notification:** A text area containing the placeholder '{!textTemplate}'.
- Help and Documentation:** A link to 'Help Documentation'.

At the bottom right, there are 'Cancel' and 'Done' buttons.

If the message itself is fairly simple, we recommend just typing it into the Textarea directly. If the message is a bit more complicated, or uses more than a couple of Resources, we recommend using a Text Template (as pictured above).

A Text Template is a great way to build a slightly more complicated Message.

- From the Flow's Toolbox, click **New Resource**
- Choose Resource Type: **Text Template**
- Provide an API name that you will use inside the Apex Action
- (optional) Add a description for this message
- Body: Make sure to choose **View as Plain Text** prior to making the message. Rich Text uses HTML for formatting, which is not **Markdown** compatible.
- Type in the text (optionally using **Markdown** for formatting) and use the Resource Picker when needed to add fields & data available to your Flow.

New Resource

* Resource Type

Text Template

* API Name

textTemplate

Description

A combination of Markdown and \$Record variables to a simple text message.
Be sure to use "View as Plain Text" when putting this together.

* Body ⓘ

Resource Picker

Insert a resource...

View as Plain Text ▾

New Lead Alert (!\$Record.Name)'s details can be visited by going to their [Lead Record]

(%%ORG_URL%%/(!\$Record.Id))

Cancel

Done

Channel - Structured Notification

Structured notifications are best used when you have specific details you want to highlight, and when you want to include Action buttons (which are just web links).

This format also allows you to have a more elaborate notification without needing to get into Adaptive Card designs.

Some examples include:

- Opportunity Closed Won Notifications
- New Lead Notifications
- Case Escalations
- Record Approvals

Permissions Required

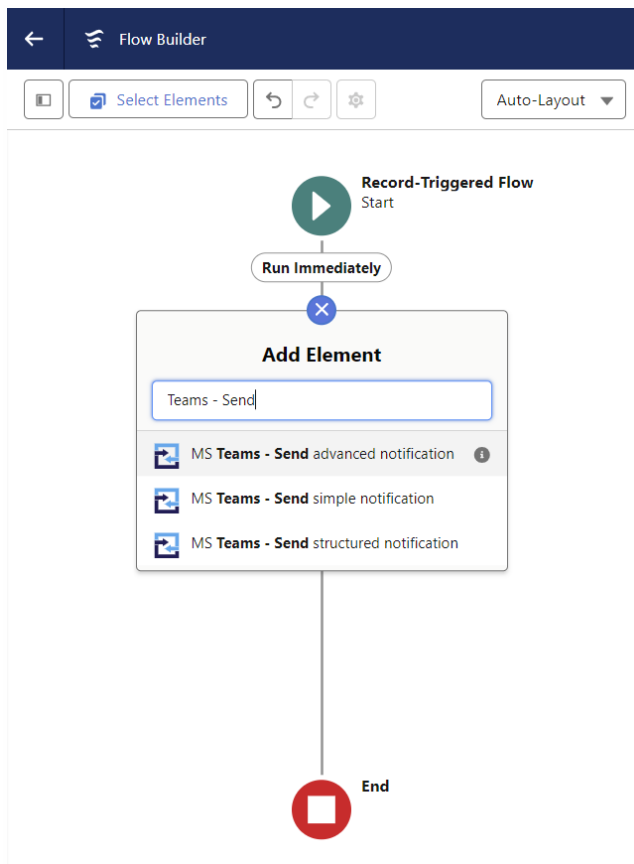
Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

Note: Notifications can only be sent to Standard Channels.

From your Flow, we will start by adding a new Element.

Pg. 26

- Search for Teams, select MS Teams - Send Structured Notification



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Notification.
- Search for the Teams Channel you want this notification to appear in.
- Provide a **Title** and **Message**.
- (optional) Add Fact Titles and Values in Pairs. These are the details that will appear in the Notification
- (optional) Add Action Labels and URLs in pairs. These are the Buttons that will appear in the notification.

Reference Send a Simple Notification for examples of using **Markdown** and **Text Templates** when necessary. This approach can be used for Title, Message, Fact Title and Labels.

New Apex Action

* Label

Send Closed Won Notification

* API Name

Send_Closed_Won_Notification

Description

This example sends a Closed Won Notification to Teams for an Opportunity

Teams - Structured Notification Information

* Channel 



Notifications



Lookup Channel

* Title 

{!\$Record.Name} - Opportunity Closed Won Alert

* Message 

Time to Celebrate! This Big Deal was won by {!\$Get_Owner.Name}

Fact Title 1

Account Name

Fact Value 1

{!\$Record.Account.Name}

Fact Title 4

Days to Close

Fact Value 4

{!\$Record.Days_to_close__

Fact Title 2

Expected Revenue

Fact Value 2

{!\$Record.Amount}

Fact Title 5

Lead Source

Fact Value 5

{!\$Record.Lead_Source__c

Fact Title 3

Commission

Fact Value 3

{!\$Record.Commission__c}

Fact Title 6

Tracking Number

Fact Value 6

{!\$Record.Tracking_Numb

Action Label 1

View Opportunity

Action Label 2

View Account

Action Label 3

View SAP Order

Action URL 1

%%ORG_URL%%/{!\$Reco

Action URL 2

%%ORG_URL%%/{!\$Reco

Action URL 3

https://myacme.sapbydes

Help and Documentation

For more information on how to use this component, please visit the [Help Documentation](#)

Cancel

Done

Channel - Advanced Notification

Advanced notifications are best when our 2 pre-configured notifications just won't get the job done.

This format also allows you to have a more elaborate notification without needing to get into Adaptive Card designs.

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

Note: Notifications can only be sent to Standard Channels.

Create your Design

Get familiar with the possibilities of Teams Notifications by using the Microsoft Adaptive Card Designer.

It can be a bit intimidating on your first visit, a few key things to absorb:

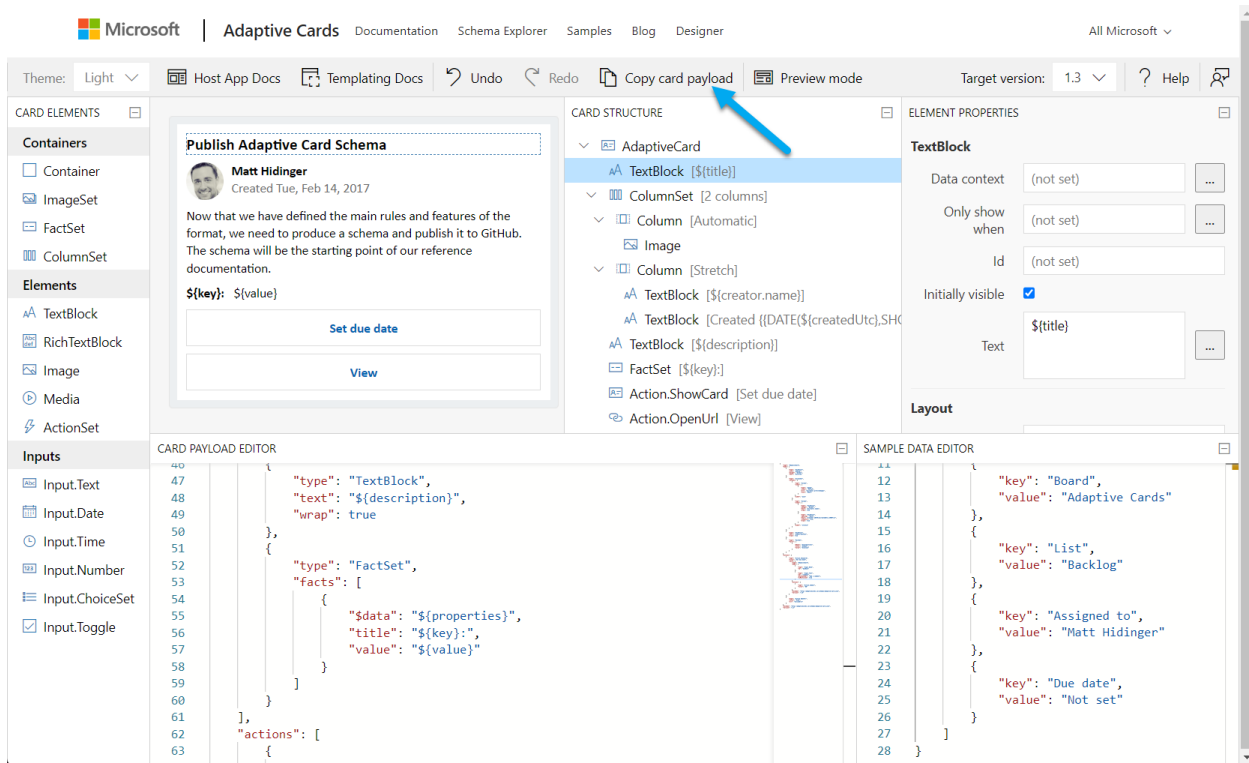
1. The toolbox of elements is found on the left, and can be dragged onto the canvas / card as needed.
2. The Card Payload Editor is a JSON view of the structure of the card. You can make changes to the JSON which is reflected in the canvas, giving you the capability to code in combination with drag & drop to build quickly.
3. The Sample Data Editor provides some information that is pulled in behind the scenes by the Card Payload Editor. This helps to separate the data from the structure.

The screenshot shows the Microsoft Adaptive Cards Designer interface. The top navigation bar includes the Microsoft logo, 'Adaptive Cards', and links to 'Documentation', 'Schema Explorer', 'Samples', 'Blog', and 'Designer'. The interface is divided into several sections:

- Left Sidebar (CARD ELEMENTS):** Contains 'Containers' (Container, ImageSet, FactSet, ColumnSet), 'Elements' (TextBlock, RichTextBlock, Image, Media, ActionSet), and 'Inputs' (Input.Text, Input.Date, Input.Time, Input.Number, Input.ChoiceSet, Input.Toggle).
- Central Canvas:** Displays a preview of an Adaptive Card titled 'Publish Adaptive Card Schema' by Matt Hidingier. The card has a 'Set due date' button and a 'View' button.
- CARD PAYLOAD EDITOR (Bottom Left):** Shows the JSON structure of the card. It includes a 'TextBlock' with a title, a 'ColumnSet' with two columns, and an 'Image' element. A blue arrow points to this section.
- SAMPLE DATA EDITOR (Bottom Right):** Shows sample data for the card. It includes a 'title' of 'Publish Adaptive Card Schema', a 'description' of 'Now that we have defined the main rules and features of the format, we need to produce a schema and publish it to GitHub.', a 'creator' object with 'name' and 'profileImage', a 'createdUtc' timestamp, a 'viewUrl', and a 'properties' object with 'Board' and 'List' values. A blue arrow points to this section.

Once you have a design you like, your next step is to use it in Flow.

- Begin by clicking the **Copy card payload** button at the top.
- If your Card Payload Editor **does not** rely on anything in the Sample Data Editor, you can just copy the JSON from that editor.



Create your Flow

From your Flow, we will start by adding a new Text Template.

- From the Flow's Toolbox, click **New Resource**
- Choose Resource Type: **Text Template**
- Provide an API name that you will use inside the Apex Action
- (optional) Add a description for this message.
- Body: Make sure to choose **View as Plain Text** prior to making the message.
- Paste the Card Payload that you copied from earlier.
- Replace certain text and values with the Resource Picker as needed to make the content of the payload dynamic to your needs.

New Resource

* Resource Type

Text Template ▼

* API Name

cardPayload

Description

This example has the default Adaptive Card Payload, with a couple of Salesforce resources merged in to make it dynamic.

* Body ⓘ

Resource Picker

Insert a resource... 🔍

View as Plain Text ▼

```
{ "type": "AdaptiveCard",
  "body": [ {
    "type": "TextBlock", "size": "Medium", "weight": "Bolder",
    "text": "{!$Record.Name} - Opportunity Closed Won Alert",
    { "type": "ColumnSet", "columns": [ {
      "type": "Column",
      "items": [ {
        "type": "Image", "style": "Person",
        "url": "{!Get_user.FullPhotoUrl}",
```

Cancel Done

From your Flow, add a new Element.

- Search for Teams, select MS Teams - Send Advanced Notification
- Fill in the Label, allow API Name to auto-fill.
- (optional) Provide a description to help you remember the intent of this Notification.
- Search for the Teams Channel you want this notification to appear in.
- Manually type in the placeholder for your Text Template.

New Apex Action

* Label * API Name

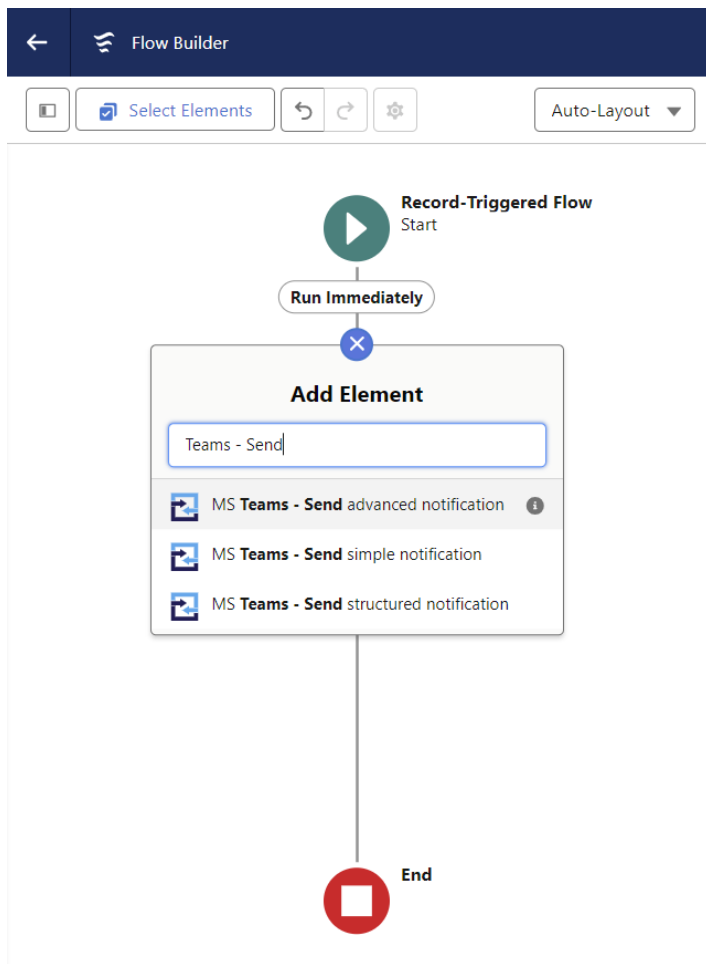
Description

Teams Advanced Notification Information

* Channel

Adaptive Card Payload

Help and Documentation
For more information on how to use this component, please visit the [Help Documentation](#)



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Notification.

- Search for the Teams Channel you want this notification to appear in.
- Provide a **Title** and **Message**.
- (optional) Add Fact Titles and Values in Pairs. These are the details that will appear in the Notification
- (optional) Add Action Labels and URLs in pairs. These are the Buttons that will appear in the notification.

Reference Send a Simple Notification for examples of using **Markdown** and **Text Templates** when necessary. This approach can be used for Title, Message, Fact Title and Labels.

New Apex Action

* Label

* API Name

Description

This example sends a Closed Won Notification to Teams for an Opportunity

Teams - Structured Notification Information

* Channel
 Notifications

☐ Lookup Channel

* Title

* Message

Fact Title 1

Fact Value 1

Fact Title 4

Fact Value 4

Fact Title 2

Fact Value 2

Fact Title 5

Fact Value 5

Fact Title 3

Fact Value 3

Fact Title 6

Fact Value 6

Action Label 1

Action Label 2

Action Label 3

Action URL 1

Action URL 2

Action URL 3

Help and Documentation

For more information on how to use this component, please visit the [Help Documentation](#)

Cancel

Done

Direct Message

Direct Messages (DMs) are ideal when you need to notify a single User.

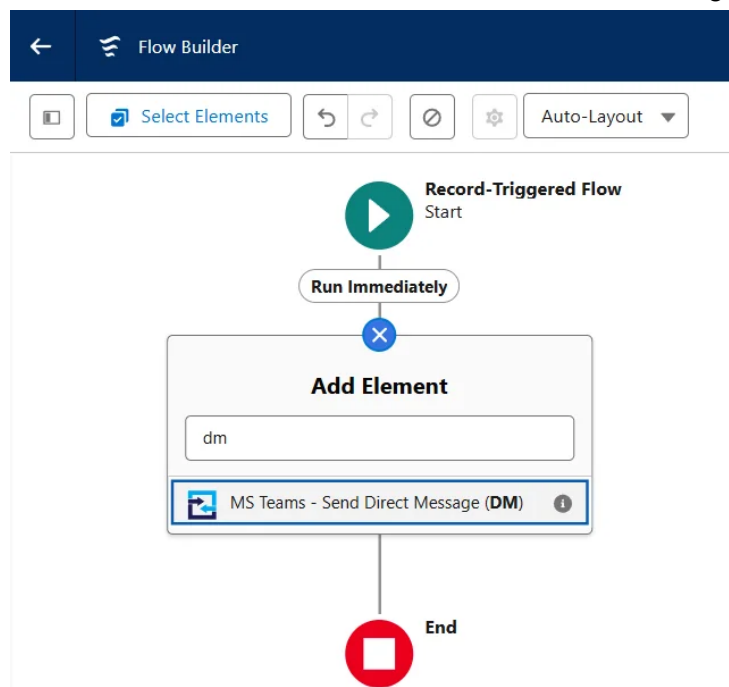
Great examples include:

- Process Action Needed Notifications
- Error Notifications
- Simple Alerts

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

From your Flow, we will start by adding a new Element.

- Search for DM, select MS Teams - Send Direct Message (DM)



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Notification.
- Search for the Teams Bot you want to send this notification through.
- Provide the recipient's Full Name or Email Address. The Teams user does **not** need to be a Salesforce User.

- Type in (or select a variable) for the Message you wish to have sent.

The screenshot shows the Salesforce Flow Builder interface. On the left, a flow diagram is visible with the following steps: 'Record-Triggered Flow Start' (green play button), 'Run Immediately' (blue circle with a plus), 'Send a DM' (blue square with a document icon, labeled 'Apex Action'), and 'End' (red square). The 'Send a DM' action is selected, and its configuration panel is open on the right.

The configuration panel for 'MS Teams - Send Direct Message (DM)' includes the following fields:

- * Label:** Send a DM
- * API Name:** Send_a_DM
- Description:** Sends a Direct Message to the Current User
- MS Teams - Send Direct Message (DM):** apex-cfl_teams__TeamsDirectMessageAction
- Teams - Direct Message Information:**
 - * Bot:** Salesforce Notifications
 - * User Email/Name:** \$User.Email
 - * Message for Notification:** A simple message can go here, and we highly recommend using a plain-text-template variable
 - Publish Event upon Notification:** No (toggle switch)
- Help and Documentation:** For more information on how to use this component, please visit the [Help Documentation](#)

- Click Done to save this Action to the Flow designer.
- Don't forget to Save & Activate the Flow when you are ready!

The message can be in Plain Text, or it can be composed using Markdown.

If the message itself is fairly simple, we recommend just typing it into the Textarea directly. If the message is a bit more complicated, or uses more than a couple of Resources, we recommend using a Text Template.

Group Chat

Group chat notifications are ideal when you need to notify multiple Users without using a Channel.

Great examples include:

- Process Status Notifications
- Error Notifications
- Simple Alerts

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

From your Flow, we will start by adding a new Element.

Create Group Chat

In order for our bot to send a notification into a Group Chat, that Group Chat must be created by our bot. When the Group Chat is created, Microsoft Teams provides us with an ID that can be used for sending future notifications.

This action performs a Callout, so if you are wanting to do this as part of a Record Triggered Flow, be sure to include an Asynchronous path.

From a Flow, we can start by adding a new Variable to hold the list of Users that will be participating in this Group Chat (they must exist in Teams).

- Open the Toolbox
- Click New Resource
- Choose Resource Type: Variable
- Provide an API name that you will use inside the Create Group Chat action
- (optional) Add a description for this variable
- Choose Data Type: Text
- Check the box for Allow multiple values (collection)
- Click Done

New Resource

* Resource Type

Variable ▼

* API Name ⓘ

varGroupMembers

Description

A list of the Teams Users that will be in the group chat.

* Data Type

Text ▼ ☒ Allow multiple values (collection) ⓘ

Availability Outside the Flow

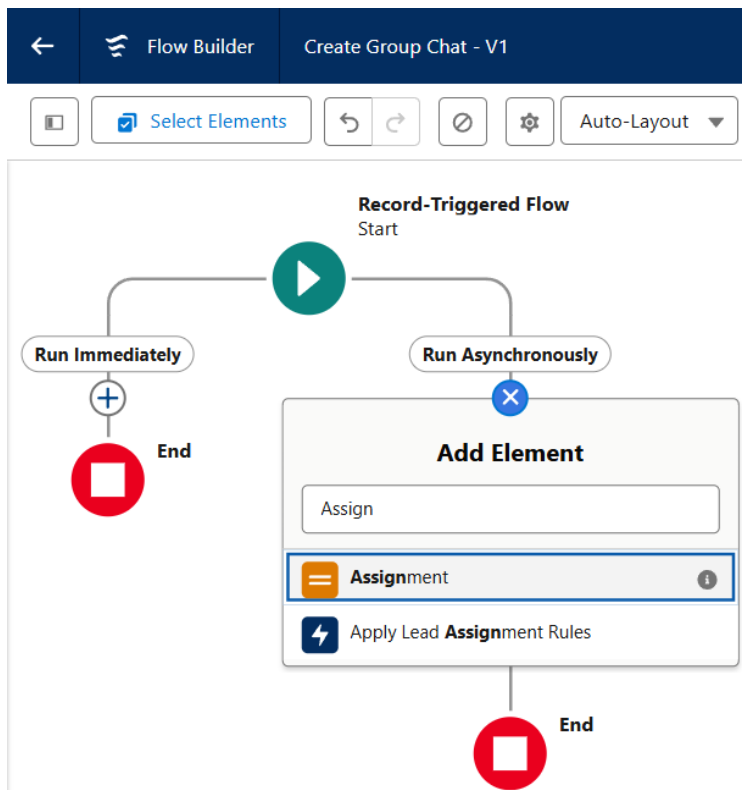
☐ Available for input

☐ Available for output

Cancel Done

Next, we need to populate the Collection of Users. There are many ways to accomplish this (depending on your data structure). In our example, we will "hard code" a list of Users.

- On the Flow Canvas, add a new Element
- Search for Assign, select Assignment



- Add Teams Users to the Collection variable.
- Close the Element's Details panel when done.

While using the Full Name can work, it is best to use Email Address in the event you have multiple people with the same name.

Assignment

* Label

Collect Users to Add to Chat

* API Name ⓘ

Collect_Users_to_Add_to_Chat

Description

Set Variable Values

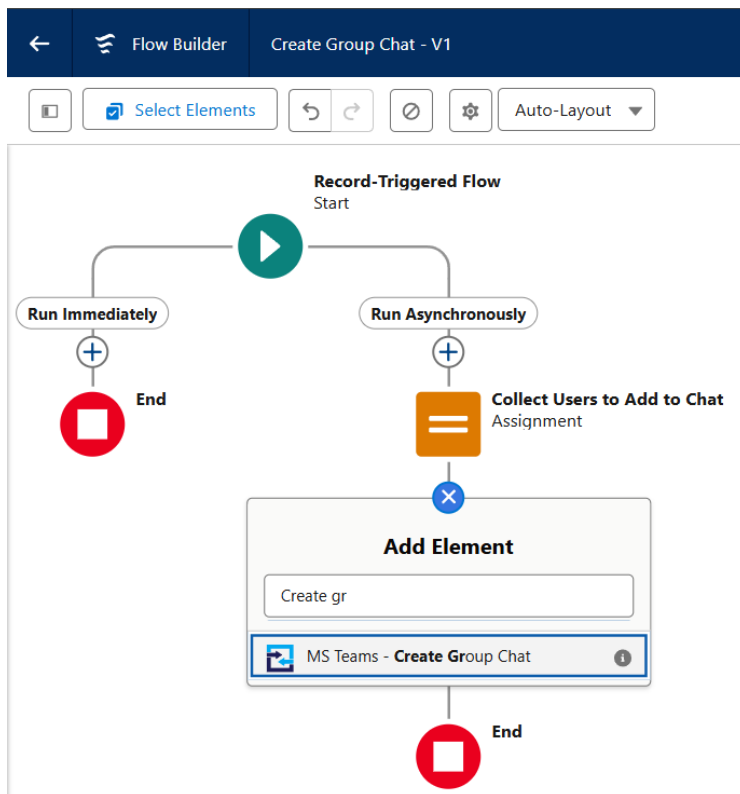
Each variable is modified by the operator and value combination.

Variable	Operator	Value
varGroupMembers	Add	Joni Sherman
varGroupMembers	Add	Patti Fernandez
varGroupMembers	Add	Alex Wilber

+ Add Assignment

Next, we can add the Element to create the chat.

- Search for Create Group, select MS Teams - Create Group Chat



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Chat.
- Search for the Teams Bot that will create the Chat.
- Pick the collection variable previously created.
- Specify the Topic to be used when naming the Group Chat.
- (optional) If you want the Bot to send notifications into this Chat in the future, enable the toggle to include the Bot as a member in the Chat.

MS Teams - Create Group Chat

* Label: Create Group Chat

* API Name: Create_Group_Chat

Description: Creates the Group Chat with all Members

MS Teams - Create Group Chat
apex-clf_teams__CreateGroupChatAction

Teams - Group Chat Information

* Bot: Salesforce Notifications

* User Email/Names: varGroupMembers

* Topic for Chat: \$Record.Name

Include Bot in Chat, necessary for sending notifications by Flow: ☒

Help and Documentation: For more information on how to use this component, please visit the [Help Documentation](#)

One of the outputs of this Create Group Chat action is the ID of the Group Chat.

Store this ID somewhere so that it can be referenced later for sending messages into the chat or retrieving messages from the chat.

This could be a Custom Field on the Record, a Custom Setting, Custom Metadata Type, or even a Label.

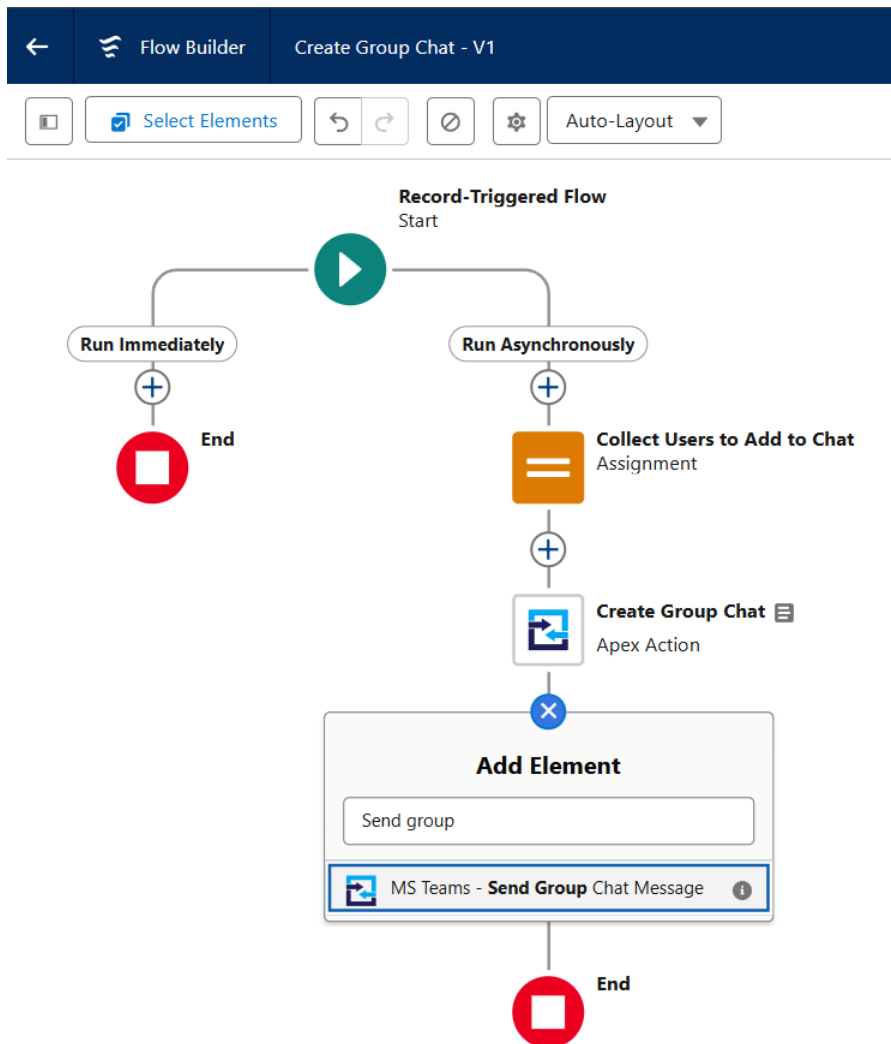
Sending the Notification

Once the Bot has created the Group Chat, and assuming the Bot was included as a Member we can now send a message.

This example assumes we have created the Notification in the same Flow as where we send the message, but as mentioned above the Chat Id can come from elsewhere.

From your Flow, add a new Element.

- Search for Send Group, select MS Teams - Send Group Chat Message



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Notification.
- Search for the Teams Bot that will send this notification. It should be the same Bot that created the Group Chat.
- Specify the Message you wish to have displayed.

- Save the Flow

The message can be in Plain Text, or it can be composed using Markdown.


MS Teams - Send Group Chat Message
×

* Label

Send intro message

* API Name ⓘ

Send_intro_message

Description

Sends the first message into the Chat, guiding recipients on what is expected to be discussed.


MS Teams - Send Group Chat Message ⓘ
 apex-cfl_teams__TeamsGroupChatSimpleMessageAction

Teams - Group Chat Message Information

* Bot ⓘ


 Salesforce Notifications
 ×

* Group Chat Id ⓘ

Create_Group_Chat.id ×

* Message for Notification ⓘ

textTemplate ×

Publish Event upon Notification ☐ No

Help and Documentation

For more information on how to use this component, please visit the [Help Documentation](#)

If the message itself is fairly simple, we recommend just typing it into the Textarea directly. If the message is a bit more complicated, or uses more than a couple of Resources, we recommend using a Text Template (as pictured above).

Merge Variables

There are a couple of merge variables available to make it just a bit easier when you create your notifications.

There are simple tokens, and the ability to @mention Teams Users and Tags.

Tokens

Variable Name	Description
%%USER_NAME%%	The Name of the current user. eg: John Doe
%%USER_EMAIL%%	The Email of the current user.
%%USER_ID%%	The 18 character Salesforce Userid of the current user. eg: 005Rt000006JpD5IAK
%%ORG_NAME%%	The name of the Sandbox, or "Production"
%%ORG_ID%%	The 18 character Id of the Org
%%ORG_URL%%	The URL for the Lightning User Interface, ideal for use with the record id. eg: https://acme.lightning.force.com/
%%CONTENT_URL%%	The URL for content stored in the org, such as files. eg: https://acme.file.force.com/

Mention Users and Tags

Mentions require a few optional Microsoft Permissions granted by the Teams admin.

Variable Name	Description
%@u:<Email>%%	Mention Teams User by Email Address. eg: %@u:john.doe@acme.com%%
%@u:<Display Name>%%	Mention Teams User by Display Name. eg: %@u:John Doe%%
%@t:<Tag Name>%%	Mention a Team's Tag by Name. eg: %@t:Marketing Ops%%

Examples

You can also use Flow resources just like you would in a Text Template...except that you have to manually type in the name of the variable.

Here are some common examples of combining the various approaches.

Example	Description
%%ORG_URL%%/{!Record.Id}	Provides a URL to the Triggering Record in a Record-Triggered Flow.
%@u:%%USER_EMAIL%%%	@ mention current User (assumes emails match in Salesforce and Teams)

Formatting with Markdown

Markdown is a lightweight markup language for creating formatted text using a plain-text editor.

Many applications have their own flavor of Markdown, and Microsoft Teams is no different.

Below you will find details on how you can use Markdown within your Notification Actions.

Available Formatting

Although Microsoft Teams has a small subset of Markdown capabilities, it should be more than enough for you to get your Notifications formatted the way you want.

Format	Syntax	How it Looks
Bold Text	**bold text**	bold text
Italic Text	<i>*italic text*</i>	<i>italic text</i>
Hyperlink	[Text](URL)	View Opportunity
Header Text	## Header Text	Header Text
Numbered List	<number> First List Item <number> Second List Item	1. First List Item 2. Second List Item
Bulleted List	* First Item or - First Item	• First Item • Second Item
Block Quote	> Text	

Examples

You can combine Markdown formatting with Flow Resource variables and our Merge variables for some interesting end results.

Need	Syntax
Hyperlink to Record (assumes Record-Triggered Flow)	[View Opportunity](%%ORG_URL%%/{!Record.Id})

Capturing Message Id

All of our Send Message actions are processed asynchronously, preventing the action from providing a return value.

To handle this, the actions all emit a Platform Event that can be handled for storing the Id where you need it.

Set Intent to Publish Event

When sending a Notification to Teams, you'll have the option of Publish Event upon Notification.

- Toggle the option to Yes.

Teams - Simple Notification Information

* Channel ⓘ

 Notifications ×

☐

Lookup
Channel

* Message for Notification ⓘ

txtMessage ×

Publish Event upon Notification

☐

No



- Set an Id to be associated with this event. Usually a Record Id will work, but you can use any Id that will help you take the right action later.
- Provide a payload that can be used later. For simple cases a single word will be enough to differentiate between different events, but you could get fancy and send whatever you like (in text form, 32k characters)
- Save the changes to this Flow Element.

Teams - Simple Notification Information

* Channel ⓘ

 Notifications ✕

☐

Lookup Channel

* Message for Notification ⓘ

txtMessage ✕

Publish Event upon Notification

☒

Yes

* Associated Record Id ⓘ

\$Record.Id ✕

Payload for Later Use ⓘ

 newAccountNotification ✕

Now that the app is publishing events when your Notifications are being sent, we need to handle them.

Handling the Published Events

Platform Event Triggered Flows don't come with a filtering capability like Record Triggered Flows do, so you'll likely just have 1 Flow that handles all notification scenarios.

This is where the Additional Payload will help out, allowing you to take different actions as necessary.

To get Started, create a new Flow.

- Select Start From Scratch

Select how you'd like to start building your automation.



Start From Scratch

Select your automation type and start building on an empty canvas.



- For the Type of Flow, select Platform Event - Triggered Flow

Select Type

Recommended

 Screen Flow Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.	 Record-Triggered Flow Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.
 Schedule-Triggered Flow Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.	 <u>Platform Event—Triggered Flow</u> Launches when a platform event message is received. This autolaunched flow runs in the background.
 Autolaunched Flow (No Trigger) Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.	 Record-Triggered Orchestration Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

- For Platform Event, select Teams Bot Conversation Created



Choose Platform Event

The flow subscribes to the specified platform event. When a platform event

* Platform Event

Teams Bot Conversation Created

> Advanced Settings

- Next, add a Decision Element. Use the Additional Payload field value to define your outcomes.

Flow Builder: Update Oppty After Msg Sent - V1

Last saved on 11/28/2024, 01:17 AM Inactive Run Save As New Version Save Activate

Select Elements

Platform Event—Triggered Flow Start

What Kind of Message Sent? Decision

New Opportunity Notifi...

Set Message Id on Opportunity Update Records

Unhandled

End

Decision

*Label: What Kind of Message Sent? *API Name: What_Kind_of_Message_Sent

Description

Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER +

OUTCOME DETAILS

*Label: New Opportunity Notification *Outcome API Name: New_Opportunity_Notification

Condition Requirements to Execute Outcome: All Conditions Are Met (AND)

Resource: Aa ...ted_e > User Defined Payload Operator: Equals Value: newAccountNotification

+ Add Condition

- Next, add an Update Records Element.
- Specify the Object type
- Use the Associated Id value to find the record (likely by Id)
- Stamp the Activity Id (this is the unique Id for the Message) into a Field of your choice on the Record.

Flow Builder: Update Oppty After Msg Sent - V1

Last saved on 11/28/2024, 01:17 AM Inactive Run Save As New Version Save Activate

Select Elements

Platform Event—Triggered Flow Start

What Kind of Message Sent? Decision

New Opportunity Notifi...

Set Message Id on Opportunity Update Records

Unhandled

End

Update Records

Set Message Id on Opportunity Set_Message_Id_on_Opportunity

Description

*How to Find Records to Update and Set Their Values

☐ Use the IDs and all field values from a record or record collection

☒ Specify conditions to identify records, and set fields individually

Update Records of This Object Type

*Object: Opportunity

Filter Opportunity Records

Condition Requirements to Update Records: All Conditions Are Met (AND)

Field: Id Operator: Equals Value: Aa \$Record > Associated Id

+ Add Condition

Set Field Values for the Opportunity Records

Field: Teams_Message_Id__c Value: Aa \$Record > Activity Id

+ Add Field

Don't forget to Save & Activate the Flow when you are ready!

Working with Channels

Create Channel Action

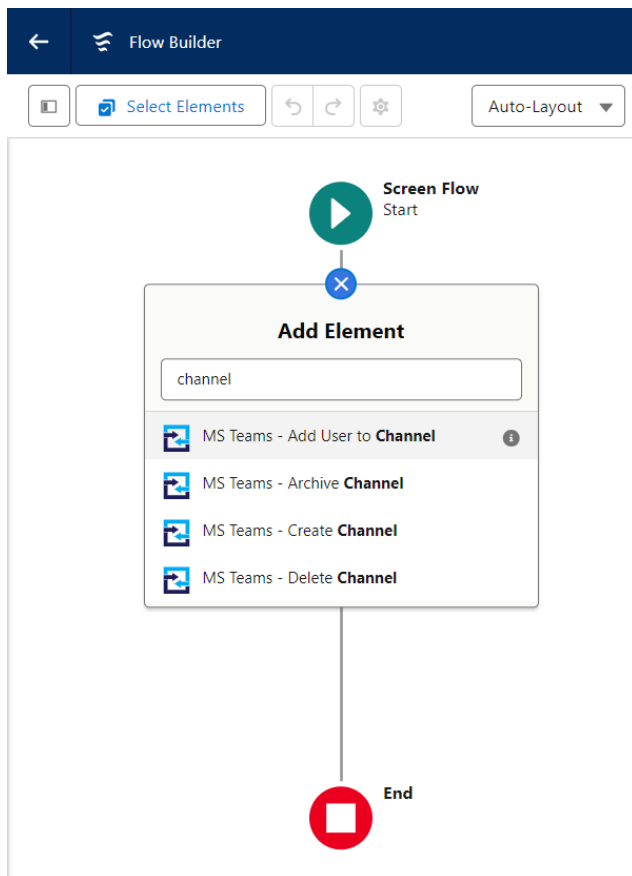
Teams channels are a great way to keep conversations organized. Using this Action, you can create Channels on demand.

Permissions Required

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.
Teams Notification Channel CRED (Perm Set)	For Users who execute the Flow, allowing them to Create, Archive, Delete Channels.

From your Flow, we will start by adding a new Element.

- Search for Teams, select MS Teams - Create Channel.



- Give the new Flow element a Label, allow the API Name to auto populate.
- (optional) Provide a description to help you remember the intent of the Channel creation.
- Search for the Team you want the Channel to be in.
- Provide a Name for the Channel. Likely you will want to include variables from a Record so that the name is unique each time the Flow executes.
- (optional) Provide a Channel Description.
- Select the Visibility for the Channel within Teams.
- Specify the Channel's owner, either by Email or Display Name.
- Click **Done** to save this Action to the Flow designer.
- Don't forget to Save & Activate the Flow when you are ready!

New Apex Action

* Label

Create Channel

* API Name

Create_Channel

Description

Create Channel for Case

Teams - Create Channel Information

* Team ⓘ

U.S. Sales

×

* Channel Name ⓘ

Opportunity - {!\$GetOpportunity.Name}

Channel Description ⓘ

Discussion for a promising Opportunity with {!\$GetAccountName}

Visibility

Standard ▼

* Owner Email/Name ⓘ

{!\$GetUser.Email}

Help and Documentation

For more information on how to use this component, please visit the [Help Documentation](#)

Cancel

Done

The Action has 1 Output variable, which is the Id of the Channel. This Id will be required for adding additional Users, as well as for Archiving/Deleting it when no longer needed. You may want to store this Id in a custom field somewhere.

Add User to Channel Action

Channels work best when there are more than 1 person in them!

This action can be used on **Shared** and **Private** channels only, **Standard** channels are shared with the Team automatically.

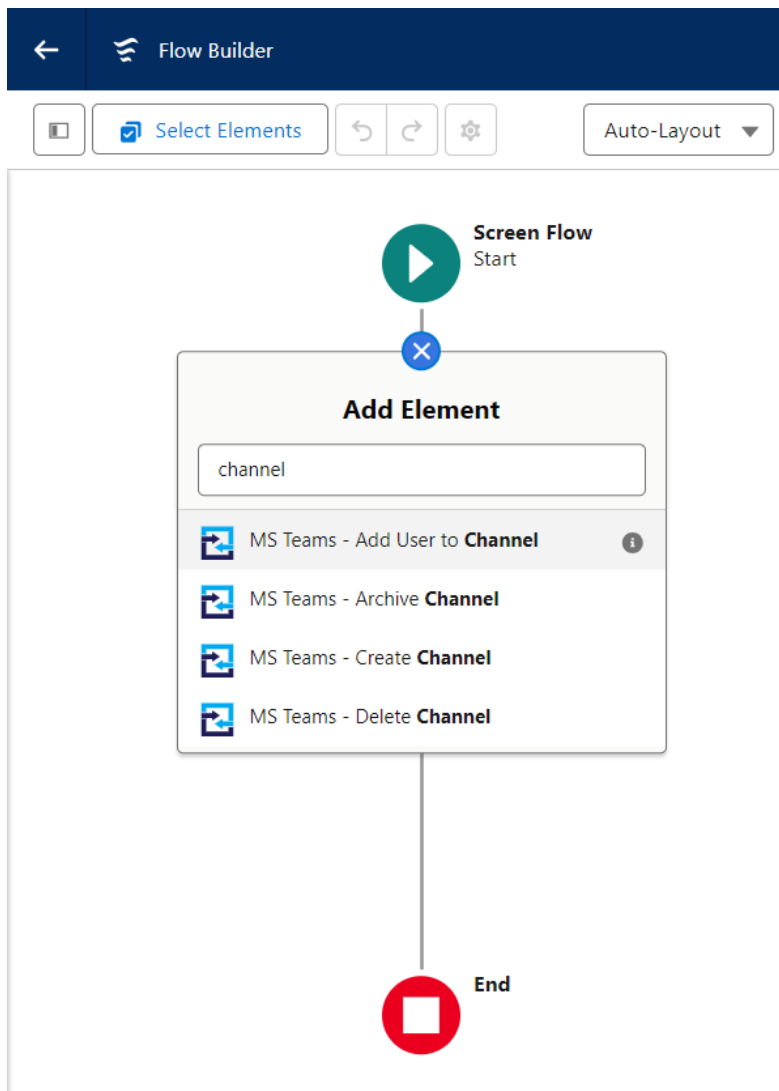
Permissions Required

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

Note: Users can only be added to Shared and Private Channels.

From your Flow, we will start by adding a new Element.

- Search for Teams, select MS Teams - Add User to Channel.



- Give the new Flow element a Label, allow the API Name to auto populate.
- (optional) Provide a description to help you remember the intent of this Action.
- Either select the Channel for a fixed value, or switch to Manual Mode and provide the Team and a variable for the Channel Id.
- Specify the User you want to add, either by Email or Display Name.
- Select the Role that this User will have in the Channel.
- Click **Done** to save this Action to the Flow designer.
- Don't forget to Save & Activate the Flow when you are ready!

New Apex Action

* Label

Add Another User to Channel

* API Name

Add_Another_User_to_Channel

Description

Teams - Add User to Channel Information

Note: this action should only be used for Private and Shared channels.

* Team ⓘ

 U.S. Sales ×



* Teams Channel Id ⓘ

{!Create_Channel.id}

* User Email/Name ⓘ

{!User_From_Loop.Email}

Role

Member ▼

Help and Documentation

For more information on how to use this component, please visit the [Help Documentation](#)

> Advanced

Cancel

Done

Archive Channel Action

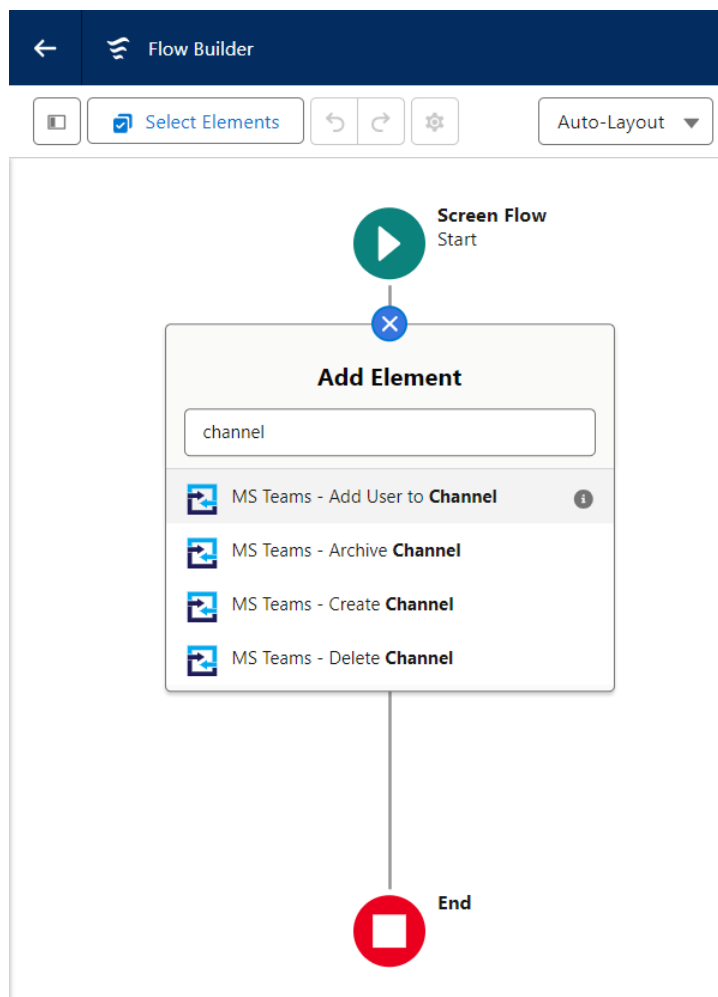
Now that your short-term use Channel is no longer needed, you may choose to Archive it so that the discussion can be kept.

Permissions Required

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.
Teams Notification Channel CRED (Perm Set)	For Users who execute the Flow, allowing them to Create, Archive, Delete Channels.

From your Flow, we will start by adding a new Element.

- Search for Teams, select MS Teams - Archive Channel.



- Give the new Flow element a Label, allow the API Name to auto populate.
- (optional) Provide a description to help you remember the intent of the Channel creation.
- Search for the Team where Channel was originally created.
- Provide the Id of the Teams Channel that will be archived. This is likely going to be a Custom Field on the Object/Record driving the archival.
- Click **Done** to save this Action to the Flow designer.
- Don't forget to Save & Activate the Flow when you are ready!

New Apex Action

* Label

Archive Channel

* API Name

Archive_Channel

Description

Archives the Channel once the Opportunity is Closed Won

Teams - Archive Channel Information

* Team ⓘ



U.S. Sales



* Channel Id ⓘ

{!\$Record.Teams_Channel_Id__c}

Help and Documentation

For more information on how to use this component, please visit the [Help Documentation](#)

Cancel

Done

Delete Channel Action

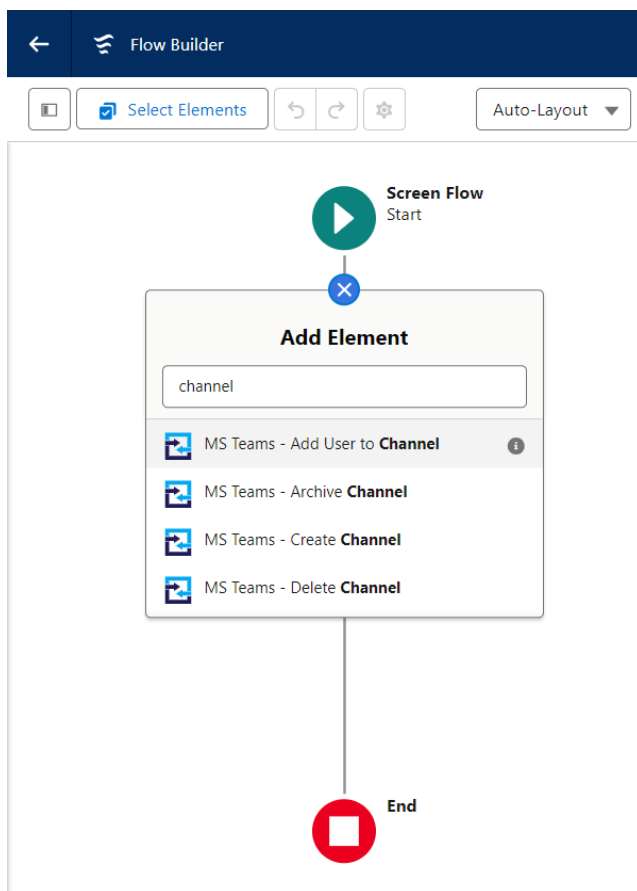
Now that your short-term use Channel is no longer needed, you may choose to permanently delete it, erasing the discussion within.

Permissions Required

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.
Teams Notification Channel CRED (Perm Set)	For Users who execute the Flow, allowing them to Create, Archive, Delete Channels.

From your Flow, we will start by adding a new Element.

- Search for Teams, select MS Teams - Delete Channel.



- Give the new Flow element a Label, allow the API Name to auto populate.

- (optional) Provide a description to help you remember the intent of the Channel creation.
- Search for the Team where Channel was originally created.
- Provide the Id of the Teams Channel that will be deleted. This is likely going to be a Custom Field on the Object/Record driving the deletion.
- Click **Done** to save this Action to the Flow designer.
- Don't forget to Save & Activate the Flow when you are ready!

New Apex Action

* Label

Delete Channel

* API Name

Delete_Channel

Description

Deletes the Channel once the Opportunity is 3 months past the Closed Won date.

Teams - Delete Channel Information

* Team ⓘ



U.S. Sales



* Channel Id ⓘ

{!\$Record.Teams_Channel_Id__c}

Help and Documentation

For more information on how to use this component, please visit the [Help Documentation](#)

Cancel

Done

Other Actions

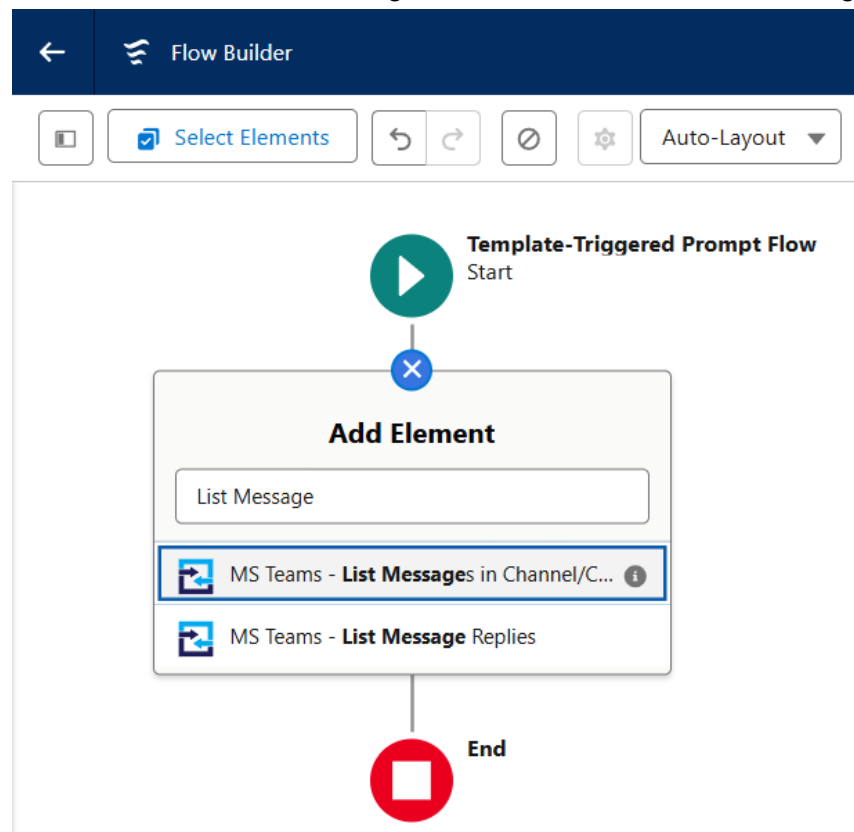
List Messages in Channel / Chat

Retrieve messages in a conversation with this action to bring the details into Salesforce. Messages can be used as-is, or provided to an LLM for further processing.

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

From your Flow, we will start by adding a new Element.

- Search for List Messages, select MS Teams - List Messages in Channel/Chat



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Action
- Search for the Teams Bot that will retrieve the messages
- Provide the Chat Id of the Group Chat / Channel you want to retrieve messages from.



MS Teams - List Messages in Channel/Chat



* Label

Get Messages for Group Chat

* API Name

Get_Messages_for_Group_Chat

Description

Retrieves messages from the Group Chat created earlier by the bot



MS Teams - List Messages in Channel/Chat

apex-cfl_teams_ListMessageAction

Teams - List Group Chat Messages Information

* Bot



Salesforce Notifications



* Group Chat Id

chatId

Concatenate into Single Text Value



Yes

Note: If you are working with a Template-Triggered Prompt Flow, you won't be able to select from \$Input. As a result, please use a Formula variable to pull the value in.

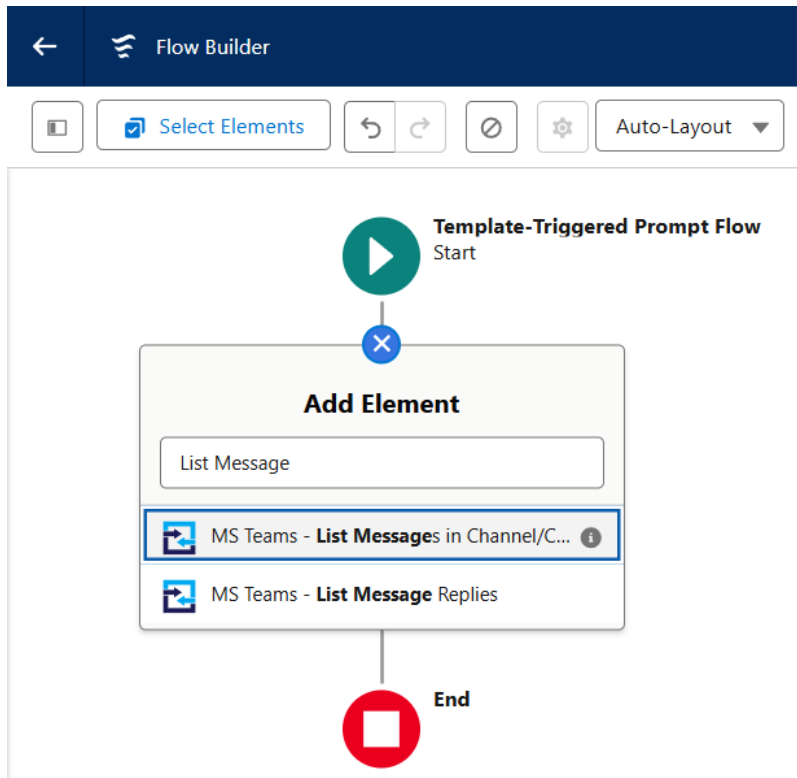
List Message Replies

Retrieve the replies from a Message with this action to bring the threaded details into Salesforce. Messages can be used as-is, or provided to an LLM for further processing.

Permission	Description
Manage Flow	To create the Flow
Teams Designer (Perm Set)	To work with this package's Flow Actions.

From your Flow, we will start by adding a new Element.

- Search for List Messages, select MS Teams - List Message Replies



- Give the new Flow element a Label, allow the API Name to auto populate
- (optional) Provide a description to help you remember the intent of this Action
- Search for the Channel (or provide it in manual mode) that has the Message.
- Provide the Message Id of the Channel's message you want to retrieve replies from.

Note: If you are working with a Template-Triggered Prompt Flow, you won't be able to select from \$Input. As a result, please use a Formula variable to pull the value in.



MS Teams - List Message Replies



* Label

Get Replies to Message

* API Name

Get_Replies_to_Message

Description



MS Teams - List Message Replies

apex-cfl_teams_ListMessageAndRepliesAction

Teams - List Message Replies Information

* Channel



Sales - Eastern



Lookup
Channel

* Message Id

messageId

Concatenate into Single Text Value



Yes

Working with Replies

You have the choice on how the replies are provided to Flow.

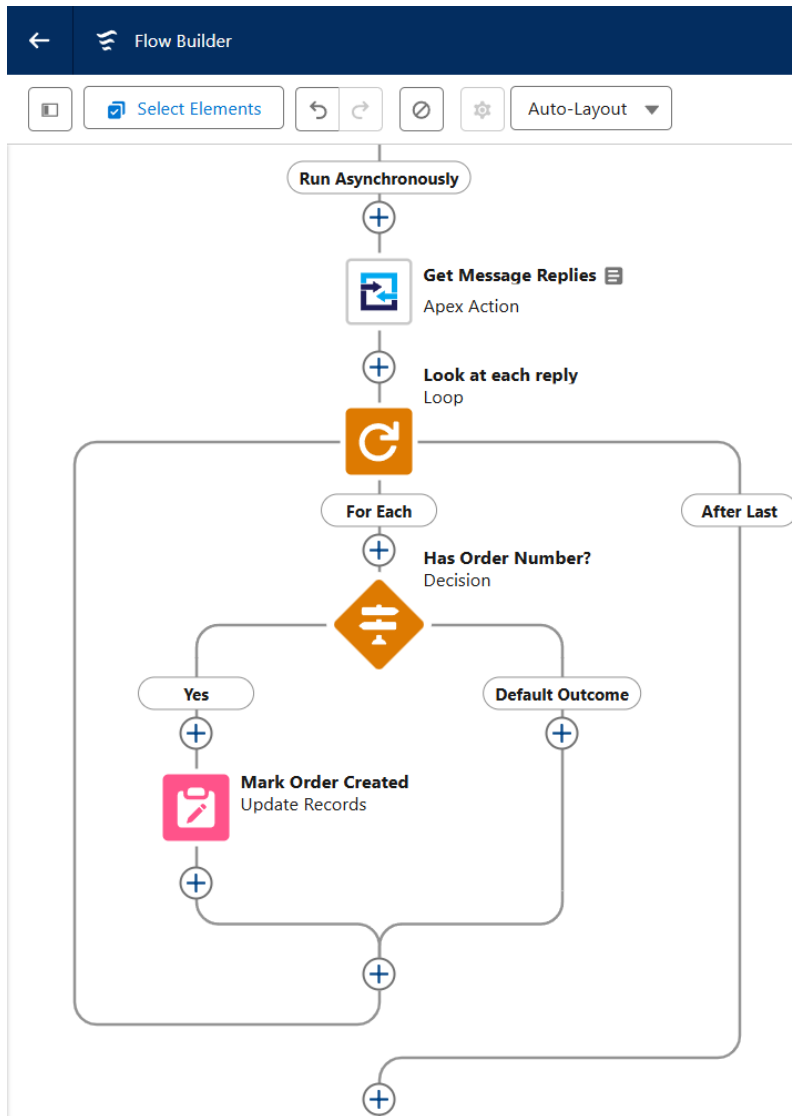
- **Individual Apex-Defined Variables:** Great for more advanced logic
- **Concatenated into Single Text Variable:** Great for passing into a Generative AI prompt.

First, let's cover the single text variable. In this format, each message is appended to the last with the sender's Name, Timestamp and message. In this scenario, be sure to use the **messagesText** output variable from the Action.

If you are working with Individual Variables, you can use a Flow Loop and gain access to each piece individually. In this scenario, be sure to use the **messages** output variable from the Action.

Only one of the 2 output variables will be populated, based on the Toggle you set during the Action creation.

In this example, our Flow retrieves the messages ([using the ID captured when the message was sent](#)) and then iterates through them looking for one we can take action on.



Within each message, the following attributes are available:

- **timestamp**: when the reply was sent.
- **displayName**: the name of the person that sent the reply
- **id**: the Id of the reply within MS Teams
- **message**: the content of the message included in the reply

* Label * API Name ⓘ

Has Order Number? Has_Order_Numt

Description

Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions :

OUTCOME ORDER ⓘ +

Yes

Default Outcome

OUTCOME DETAILS

* Label * Out ⓘ

Yes Yes

Condition Requirements to Execute Outcome

All Conditions Are Met (AND) ▼

Resource

{!Look_at_each_reply}

+ All Resources > Current Item from Loo...

Properties

When

☒ If

☐ C

timestamp ⓘ

A_a displayName

A_a id

A_a message ⓘ

+ New Resource

Operator

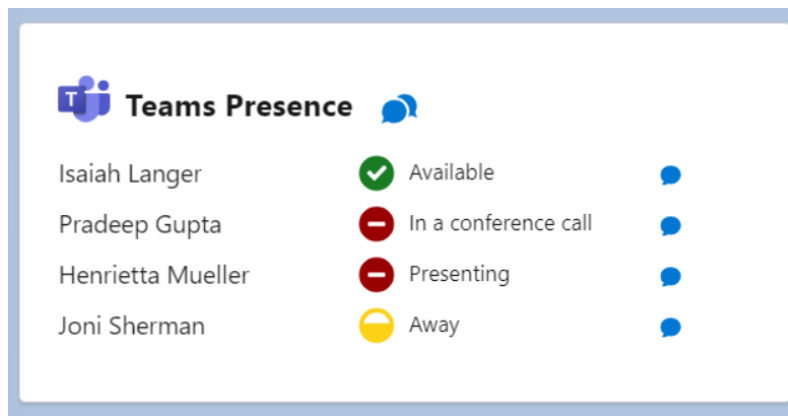
Equals

Page Components

Presence Display Component

Highlight the current status of Teams Users from within Salesforce on a Lightning Record Page.

This component can show the status for both a single User as well as a list of Users through a Record's related list (such as AccountTeam or OpportunityTeam).



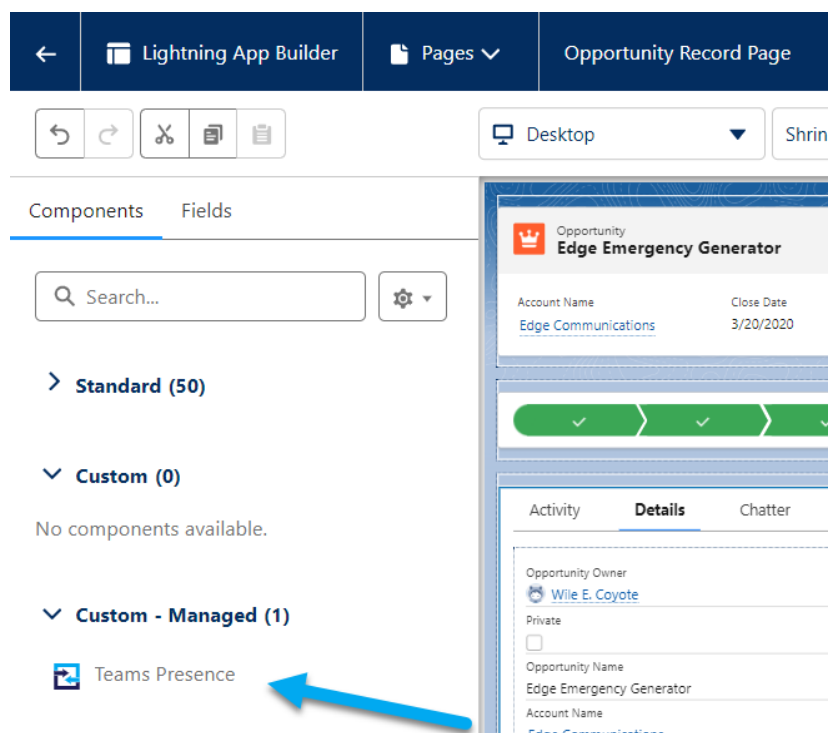
Permissions Required

Permission	Description
Customize Application	To create and save Lightning pages in the Lightning App Builder

Add Component to Page

To get started, create or edit a Lightning Record Page.

In the Components Toolbox, under the **Custom - Managed** section grab the **Teams Presence** component and drag it onto the Lightning page.



To get started, create or edit a Lightning Record Page.

In the Components Toolbox, under the **Custom - Managed** section grab the **Teams Presence** component and drag it onto the Lightning page.

Once added, a properties panel will appear on the right.

Property	Description
Component Title	Set the title of the component when it is rendered.
Single User Details	The API field name (or path) of a field that contains the Teams User's email or display name.
Related Records	The relationship that will be used to get a list of other Users to display.
User Details (multiple)	The API field name (or path) of a field that contains the Teams User's email or display name, on the Related Record.
Sort Method	How you would like to sort the list of Users, either strictly alphabetical, or first by status.
DM Button Display	Specifies if the DM icon (speech bubble) should appear for the group and/or individual.
Compact Mode	Changes the appearance to be a 2 column layout relying a bit more on icons.

Analyze

Activation...

Save

Page > Teams Presence

Component Title ⓘ

Teams Presence

Single User Details ⓘ

Owner.Email

Related Records ⓘ

User Details (multiple) ⓘ

User.Email

Sort Method ⓘ

StatusThenDisplayName

DM Button Display ⓘ

IndividualGroup

Set Component Visibility

Filters

+ Add Filter

Once the component looks good, save the page and activate it if necessary.

Presence Caching

The component updates itself every 10 seconds, and makes N+1 API calls to Microsoft Teams. 1 request to lookup each user's Teams Azure Active Directory ID, another to batch retrieve current status.

In environments where there might be a high volume of users continuously retrieving updates, the Teams API might start throwing errors (it does have limits).

You can enable Platform Cache in Setup for our application. 1MB of Org Cache will support approximately 500 looked up users. Entries will be cached for 5 minutes (the minimum time we can cache an entry).

Managing AppExchange Subscription

As an AppExchange solution, a lot of the management of the App happens from the AppExchange!

This allows you to be completely self-serve and not have to talk with another person (though we really would love to have a chat!).

Please note: Cyfuno Labs does not have access to your Invoices or Subscription.

Permissions Required

To view or make any adjustments to your AppExchange Subscription, the Salesforce User will require the **Manage Billing** permission.

Billing Details

Whether you need to update Billing Contact information or payment information, the AppExchange again is the place to go.

From **My Installs and Subscriptions**, find our app and click on the Arrow to the right. In the little pop-up menu, click **Manage Subscription**.

Your subscription details will appear. From here you can **Edit** both Contact Details as well as Payment Details.

Click **Done** when you have made the right changes.

Invoices / Receipts

The AppExchange handles all billing and Invoicing.

From **My Installs and Subscriptions**, find our app and click on the Arrow to the right. In the little pop-up menu, click **Manage Subscription**.

When you see your Subscription details, scroll down until you find **Payment History**. From there, you can find all Salesforce-generated details for your regularly occurring invoices.

Updating the Package

When it is time to update, you start by logging back into the AppExchange.

From **My Installs and Subscriptions**, find our app and click on the **Update Available** link.

salesforce appexchange

Search AppExchange

Home Explore Collections Consultants Learn

Filters ▾ | All My Installs & Subscriptions 6 Results

Don't see what you're looking for? [More Info](#)

LISTING	LISTING TY...	UPDATED	INSTALLED	PAYMENT STATUS
 Notifications in Microsoft Teams Cyfuno Labs	APP	Update Avail...	✓	Installed - Paid

You will need to agree to the Terms and Conditions, and then click **Confirm and Install**.

Search AppExchange

Terms & Conditions

Cyfuno Labs, LLC Terms & Conditions

<https://www.cyfunolabs.com/vikundi/app-tos/>

Salesforce.com Terms & Conditions

If salesforce.com is not listed as the provider of this application, or if the provider listed is Salesforce Labs, then the application is not provided or warranted by salesforce.com. You and/or other users at your company may be able to use this application to access your or your company's data in the

☒ I have read and agree to the above terms & conditions

Close Confirm and Install

Next, you can verify that the upgrade is happening, **Install it for Admins Only** and click **Upgrade**.



Upgrade teams-notifications

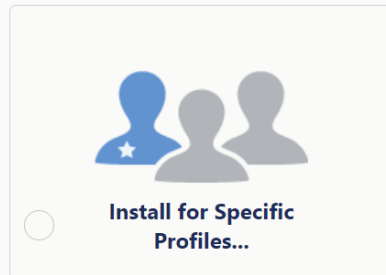
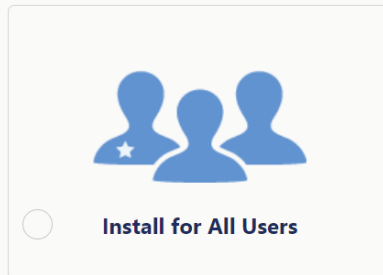
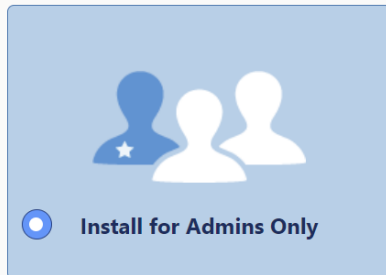
By Cyfuno Labs



An earlier version is installed. It can be upgraded while preserving the existing data.

Installed: 2.1.0 (2.1)

New Version: 2.2.0 (2.2)



Upgrade

Cancel

App Name

teams-notifications

Publisher

Cyfuno Labs

Version Name

2.2.0

Version Number

2.2

Additional Details

[View Components](#)

You will need to approve 3rd Party Access. This will allow the package to contact Microsoft's API endpoints directly from your Salesforce Org.

×

Approve Third-Party Access

This package may send or receive data from third-party websites. Make sure you trust these websites. What if you are unsure?

Website	SSL Encrypted
graph.microsoft.com	☒
login.microsoftonline.com	☒
smba.trafficmanager.net	☒

☒ Yes, grant access to these third-party web sites

Continue

Cancel

After clicking **Continue**, all you need to do is wait for the upgrade to complete.

Uninstalling the App

If you no longer need to use our solution, you can cancel your Subscription from the AppExchange.

Before performing these steps, you will need to perform a few clean up steps to allow Salesforce performing the removal successfully.

- Ask your Teams Admin to delete the Bot.
- Verify that there are no longer any Active **Teams Channels** in Salesforce.
- Remove the Site that you configured.
- Remove our Flow Actions from each Flow (and version of Flow).

From **My Installs and Subscriptions**, find our app and click on the Arrow to the right. In the little pop-up menu, click **Uninstall**.

You may be asked to login to your Org again as another verification.

Review the items that will be removed, then scroll to the bottom of the screen and complete the uninstallation.