Oracle Fusion Cloud Sales Automation

How do I integrate Microsoft Teams with Redwood User Experience?

Oracle Fusion Cloud Sales Automation
How do I integrate Microsoft Teams with Redwood User Experience?

F89626-07

Author: Oracle

Contents

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Some application pages have help icons to give you access to contextual help. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. If the page has contextual help, help icons will appear.

Get Support

You can get support at *My Oracle Support*. For accessible support, visit *Oracle Accessibility Learning and Support*.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program*. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

1 Get Started with Integrating Microsoft Teams with Redwood User Experience

Overview of Integrations with Microsoft Teams

Microsoft Teams provides functionality for collaboration enabling users to communicate with teams and channels and conduct web conferencing for seamless virtual meetings and teamwork.

Oracle provides different integration solutions with Microsoft Teams to cater for your specific sales application requirements. Collaboration via teams and channels consists of the following two solutions:

1. *Microsoft Teams Collaboration across Fusion Applications*

   This integration solution with Microsoft Teams works consistently across Fusion applications. You can create a team in Microsoft Teams for Oracle Fusion business objects, such as requisitions for Oracle Procurement and opportunities for Oracle Sales, and view Microsoft Teams conversations within the Fusion application user interface.

   For more information about how to set up this preferred solution, see the **Microsoft Teams Collaboration across Fusion Applications** section of this playbook where you'll be directed to links to implement this solution as is, or can change and extend the solution to meet your specific needs.

   > **Note:** Oracle recommends that you set up this integration as your preferred solution.

2. *Microsoft Teams Collaboration with Oracle Sales*

   Oracle provides an alternative Microsoft Teams integration solution that enables you to simply click the **Create Team in Microsoft Teams** action from the opportunity page of your Oracle Sales in the Redwood User Experience to create a dedicated team in Microsoft Teams. Once the team is created, salespeople can then use the **Go to Microsoft Teams** action to navigate to the team and collaborate with others to pursue the opportunity. With this solution, salespeople can start to collaborate directly in Microsoft Teams for the opportunity and are provided with an option to easily navigate to it.

   For more information about how to set up these steps, see the **Microsoft Teams Collaboration with Oracle Sales** section of this playbook and follow the setup steps required.

The *Microsoft Teams Integration for Web Conferencing* section of this playbook describes the steps to enable, integrate, and verify a Microsoft Teams web conference integration with your Sales in the Redwood UX. Once integrated, salespeople can start a Microsoft Teams web conference meeting or schedule a meeting from within their Oracle sales application.

2 Microsoft Teams Collaboration across Fusion Applications

About Integrating with Microsoft Teams across Fusion Applications

Oracle offers a sample solution that you can use to implement a collaboration solution integrated with Microsoft Teams. This sample solution works consistently across Fusion applications such as Oracle Procurement and Oracle Sales and allows salespeople to view conversations within their sales application interface.

This preferred solution enables you to view conversations:

- Stored in channels, within Oracle Fusion, without needing to use the Microsoft Teams client
- That support text, images, emojis and links to attachments
- In Oracle Fusion without being a member of the team.

When you navigate to the specific conversation in Microsoft Teams, then you're automatically added to the team and receive Microsoft Teams notifications.

For information about how to use Microsoft Teams for networking and collaborating across a range of internal teams, external partners, or suppliers, see *Oracle Fusion Applications Integration with Microsoft Teams*.

Note: Oracle recommends to use this preferred solution to integrate with Microsoft Teams across Fusion applications.

Prerequisites

The following applications and software are required for Oracle Fusion Applications Social Network Integration:

- Oracle Fusion Applications Release 23c (or later)
- Microsoft Teams
- Oracle Integration Cloud Service
  - Integration Service (IC)
  - Visual Builder Cloud Service (VBCS)

Where to download?

Please use the link below to download the resources including source code, architecture diagram, installation guide, user guide, and all licensing files:

- *Oracle Fusion Applications Social Network Integration*.

3 Microsoft Teams Collaboration with Oracle Sales

About Integrating Redwood UX with Microsoft Teams Using Smart Actions

As an administrator, you manage the Oracle Microsoft Teams integration for users using Sales in the Redwood UX. This integration with Microsoft Teams can be enabled via extensibility using a REST based smart action, an object function smart action and a formula field on the object to evaluate whether the team exists in Microsoft Teams for the record.

> **Note:** Oracle recommends to use its preferred solution to integrate with Microsoft Teams across Fusion applications. See *Oracle Fusion Applications Integration with Microsoft Teams* for more information.

This topic explains how to enable the Microsoft Teams integration to create a team in Microsoft Teams for an opportunity. However, you can also use these steps to create a team and access Microsoft Teams for custom objects as well.

> **Note:** The steps to setup the integration for custom objects are very similar, except when creating the formula field and object functions. The sample opportunity setup scripts provided contain **OptyId**. This value must be replaced with the ID field for the custom object in all the sample scripts.

Here are the high-level steps to set up the integration of Oracle Sales Redwood UX and Microsoft Teams:

1. Create the Microsoft Azure application.
2. Register the Microsoft Azure application in Oracle Sales
3. Register a REST Web Services connection.
4. Create a custom formula field.
5. Create an object function server script.
6. Define the **Create Team in Microsoft Teams** smart action.
7. Define the **Go to Microsoft Teams** smart action.

Before You Start

You need two users to complete the setup:

- A setup user in Sales
- An administrator user for the Microsoft Azure

For this integration to work, make sure:

- Your Oracle Sales Redwood UX supports fragments.
- That the email address in the resource record for the user using Sales for Redwood UX matches the user's email address in Microsoft Teams.

- The opportunity team owner must be a Microsoft Teams user and also be the owner of the team in Microsoft Teams.
- To add the opportunity team members to the resource team of the record before the Microsoft Teams team is created. This ensures that these resources can be automatically added to the team in Microsoft Teams when the team is created.

Enable the Archive Option for Microsoft Teams

After you've successfully created the Microsoft Azure application and registered the app in Oracle Sales, follow these steps to enable the archive option so that when an opportunity closes, the corresponding Microsoft Team gets archived automatically.

1. In the **Setup and Maintenance** work area, go to:
    - Offering: Sales
    - Functional Area: Integrations
    - Task: Manage Microsoft 365
2. On the Manage Microsoft 365 page, go to the Microsoft Teams tab.
3. Select the **Archive Opportunity Team in Microsoft Teams Enabled** checkbox.
4. Click **Save and Close**.

Create the Microsoft Azure Application

To create the Microsoft Teams add-in, you must first create the Microsoft Azure application. To get your host URL follow these steps:

1. Sign in to the Microsoft Azure portal at `https://portal.azure.com`.
2. Click **Admin** on the home page.
   The Admin center page appears.
3. Click **Microsoft Entra ID** under Azure Services.
   The Microsoft Entra ID Admin Center page appears.

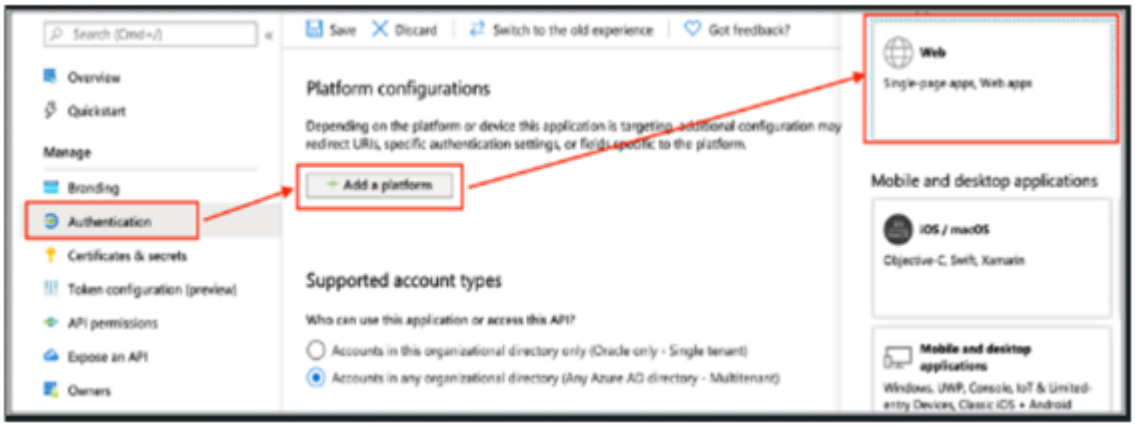
   > **Note:** If you're a first-time user, then you must sign up for your Microsoft Entra ID.

4. Click **App registrations**.
   The Applications list page appears.
    a. Click **+ New registration**.
       The Register an application dialog box appears.
    b. For Name, enter a name for the application, for example, Microsoft Teams Collaboration.
    c. Under Supported account types, select **Accounts in this organizational directory only (Oracle only - single tenant**.
    d. Leave Redirect URI (optional) blank.
    e. Click **Register** to create the Microsoft Teams add-in application.
       The application appears on the App registrations page.

The Application details page displays the Application (client) ID and Directory (tenant) ID.

> **Note:** Keep a note of these values because you insert these in the Manage Microsoft 365 task page later on. See the *Register the Microsoft Azure Application in Oracle Sales* section for more information.

5. Next click **Authentication** and from the Authentication page, click **Add a platform** and select **Web** as shown in this screenshot:



6. In the **Redirect URIs** section of the Configure Web dialog box, add the URI of your environment (for example, `https://sample-pod.oraclecloud.com`).
7. In the Implicit grant section, select **Access tokens** and **ID tokens**.
8. Click **Configure**.
9. Next click **Certificates & secrets**, and from the Add a client secret section, click **+ New client secret**.
10. Select a duration for the expiration of the Client Secret. For example, if you select **12 months**, the Client Secret becomes invalid after 12 months, and the automatic refresh stops. The sync process will start only after you sign in to the client again.

    If you select **Custom**, enter the start and end dates, which must be within 24 months. After the end date, you must sign in to the client again, to start the sync.
11. Click **Save**.

    The application generates a value for the key. Keep a note of this value because you'll use this as the Application Key in the Manage Microsoft 365 task page later on.
12. Click **API permissions** and click **Add a permission**. The Request API permissions pane appears.
13. Scroll to the end of the pane and click **Microsoft Graph Commonly used Microsoft APIs**.
14. Click **Application permissions** and select the following permissions:
    - `Directory.Read.All`
    - `Directory.ReadWrite.All`
    - `Group.Read.All`
    - `Group.ReadWrite.All`
15. Click **Add permissions**.
16. Click **Grant admin consent for Oracle** and then click **Yes** on the confirmation dialog box.

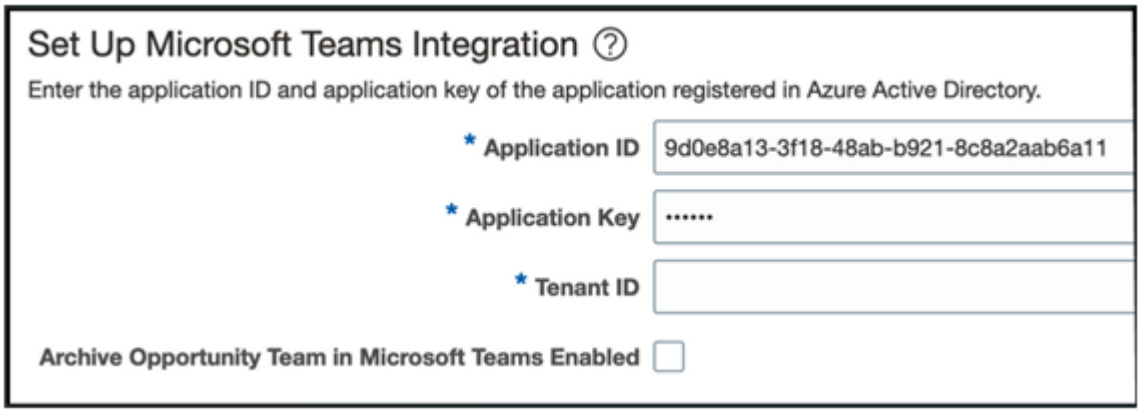
Register the Microsoft Azure Application in Oracle Sales

In Oracle Sales, use the Manage Microsoft 365 task to register the Microsoft application you previously set up in the Microsoft Azure Directory.

1. Sign in as a setup user or as the sales administrator.
2. Navigate to:
    - Offering: Sales
    - Functional Area: Integrations
    - Task: Manage Microsoft 365
3. Click the **Microsoft Teams** tab.
4. The first time you set up the integration, a usage agreement notice displays for your information:

    Usage Agreement: This Oracle-Microsoft integration or plug-in may enable you to link to or transmit your content or third-party content to, or otherwise access, or retrieve content from, Microsoft and its platforms, users, or services. Oracle doesn't control and isn't responsible for Microsoft sites or platforms or services, the performance or availability of the services, or any content received or sent. You bear any and all risks associated with access to and use of Microsoft sites, platforms, and services and are solely responsible for entering into and compliance with separate terms between you and Microsoft. Oracle isn't responsible for the security, protection or confidentiality of such content (including obligations in the Hosting and Delivery Policies and Data Processing Agreement and Oracle's Privacy Policy) that's transmitted to such Microsoft sites or platforms or services. You're solely responsible for obtaining or having any required consents or other legal basis for your use of this integration or plug-in. Oracle reserves the rights to terminate your connection or integration if your use of this integration or plug-in violates the terms of your Agreement(s) with Oracle.
5. Enter the Application ID, the Application Key, and the Tenant ID. See *Create the Microsoft Azure Application* for information about how to generate these details.

    Let's say you enter the name as Oracle Microsoft Teams Integration. When you sign in to your Microsoft 365 account, you can see Oracle Microsoft Teams Integration as an add-in that you can select. Here's a sample screenshot:



Register a REST Web Service Connection

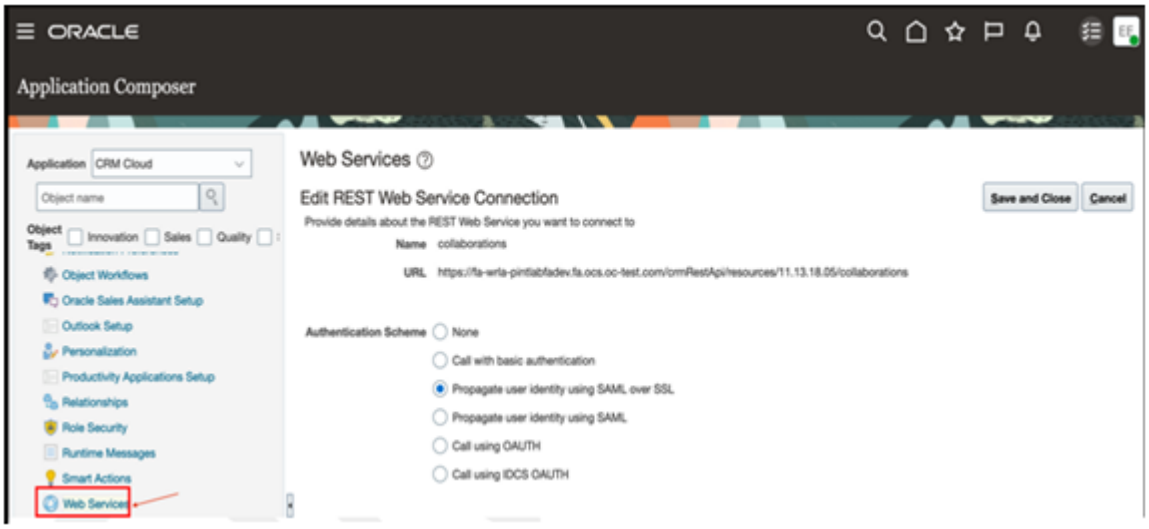
To register a web service for use in your scripts, you must first select Web Services in the Common Setup pane in Application Composer.

The collaboration API is used to determine if a Microsoft Team is already created for the opportunity for the **Create Team in Microsoft Teams** action. It is also used to retrieve the Microsoft Teams URL for the team associated with the opportunity for the **Go to Microsoft Teams** action.

Complete the following steps to register a REST Web Service Connection to use for your Sales for Redwood UX integration with Microsoft Teams.

1. Navigate to **Configuration** > **Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. In Application Composer, select **Common Setup**, and click **Web Services**.
4. Click **REST** to create a new connection. The **Edit REST Web Service Connection** page displays.
5. Provide the details about the REST Web Service to connect to as per the following:
    a. **Name**: collaborations
    b. **URL**: https://<env>/crmRestApi/resources/11.13.18.05/collaborations

    (replace <env> with your environment URL, such as http://localhost:3000-test.com/crmRestApi/resources/11.13.18.05/collaborations)



6. In the **Authentication Schema**, select **Propagate user identity using SAML over SSL**.
7. Configure the GET method as follows:
    - Request Payload: **Default (Schema URL)**
    - Response Payload: **Code Sample**

    Select the **Code Sample** option and copy the following payload code.

> **Note:** Replace **<env>** with your environment URL, such as **http://localhost:3000-test.com/crmRestApi/resources/11.13.18.05/collaborations**

```
{
 "CollabId": "300100575968676",
 "CreatedBy": "MHoope",
 "CreationDate": "2023-07-24T03:38:27.103+00:00",
 "Weburl": "https://teams.microsoft.com/l/team/19%3aH7m-L-l4hQrN-
J8Ych4c3fEAxb09AoknST_z4dV1ghY1%40thread.tacv2/conversations?groupId=e512e688- d8b8-4a5a-
b34e-8f508675e8aa&tenantId=33b71f2d-7f52-4370-bd2c-08efb530b1c3",
 "ObjectId": "300100574254353",
 "ObjectName": "opportunities",
 "Name": "OPPTY - Test Opty 000",
 "Description": "This team is specifically for the Test Opty 000 opportunity: IBM",
 "TeamRelationshipName": null,
 "links": [

 { "rel": "self",
 "href": "https://<env>/crmRestApi/resources/11.13.18.05/collaborations/300100575968676",
 "name": "collaborations",
 "kind": "item"
 },

 { "rel": "canonical",
 "href": "https://<env>/crmRestApi/resources/11.13.18.05/collaborations/300100575968676",
 "name": "collaborations", "kind": "item"
 }
 ]
}
```


8. Configure the POST request and response to use **Default (Schema URL)**.
9. Click **Save and Close**.

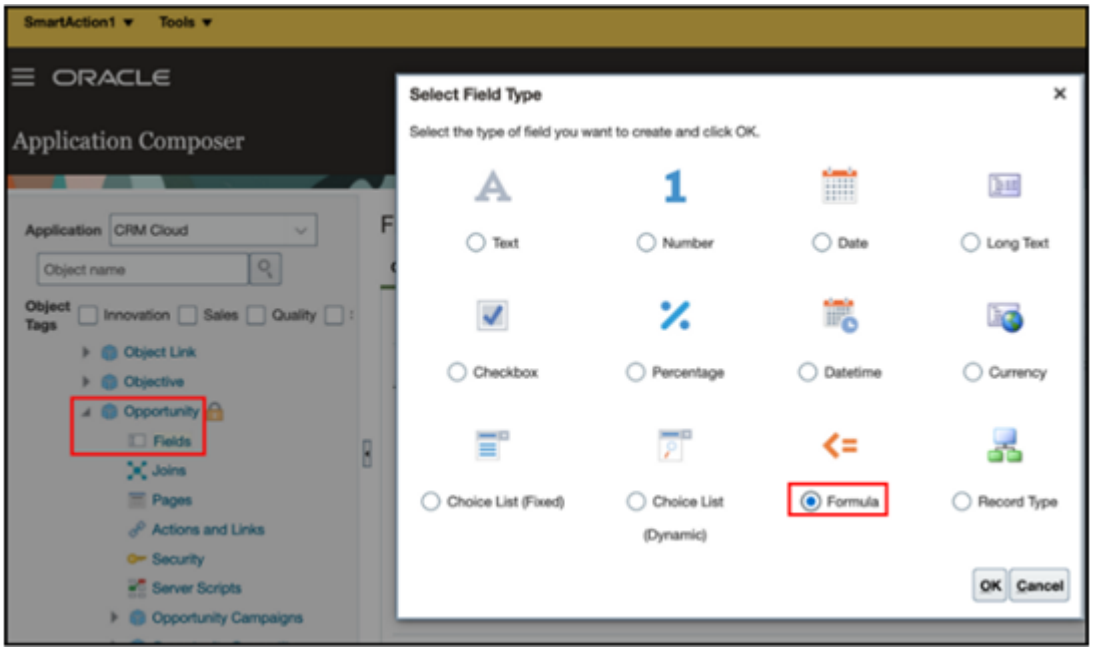
Create a Custom Formula Field

A formula field is required to identify whether a Microsoft Teams already exists for the record. To create a formula field, follow these steps:

1. Navigate to **Configuration** > **Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. In Application Composer, expand the **Standard Objects** node, and then expand **Opportunity**.

> **Note:** You can also use these steps to create a team and access Microsoft Teams for other objects such as custom objects as well.

4. Click **Fields** as shown in this example screenshot.



5. Click the **Custom Fields** tab.
6. On the Custom Fields tab, click **New**.
7. Select Formula as the field type with the following details.
    a. **Formula Type**: Text
    b. **Display Label**: microsoftTeamCreated
8. Click **Next**.
9. Copy and paste the following groovy code:

```
def conn = adf.webServices.collaborations;
try
{
 // Provide query parameter for the account object you want to receive
 def queryParams = ['finder':'GetCollaborationsForObjectFinder;ObjectId='+OptyId+',Validate=true']
 conn.dynamicQueryParams = queryParams
 def collabResponse = conn.GET()
 if (collabResponse.count == 0 ) {
 return "NO";
 }
 if (collabResponse.count > 0 ) {
 return "YES";
 }
 return "ERROR";
}
catch(Exception ex)
{
 println(conn.statusCode+"");
 println(conn.httpErrorResponse+"");
 return "ERROR";
}
```

> **Note:** To set up the integration for custom objects you can copy the scripts provided but you must replace the OptyId value with the ID field for the custom object in all the sample scripts.

10. Click **Submit**.

Create an Object Function Server Script

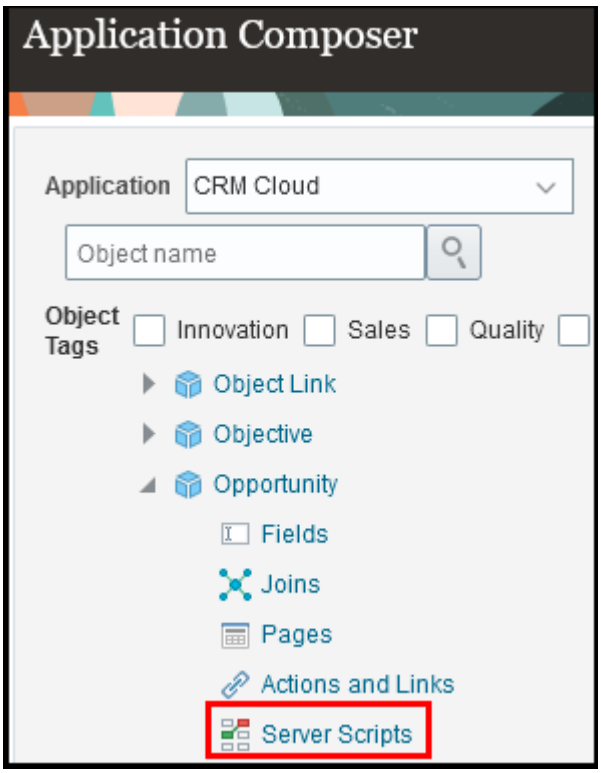
As an administrator, you must define the object function server script to be used by the Go to Microsoft Teams smart action to retrieve the Microsoft Teams URL associated with the record.

Here's how to define the GoToTeams function server script in Application Composer.

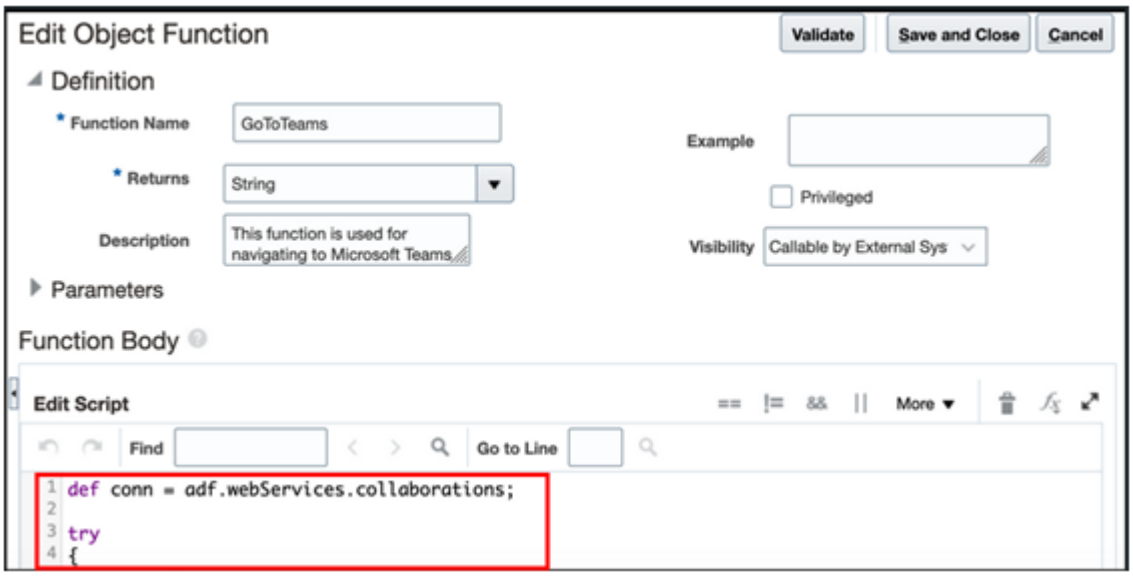
1. Navigate to **Configuration** > **Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. In Application Composer, expand the **Standard Objects** node, and then expand **Opportunity**.

> **Note:** You can also use these steps to create a team and access Microsoft Teams for other objects such as custom objects as well.

4. Click **Service Scripts** as shown in this example screenshot.



5. Click the **Object Functions** tab to display a screen similar to the following:



6. Enter required details as follows:
   - **Function Name:** GoToTeams
   - **Returns:** String
   - **Visibility:** Callable by External Systems
7. Enter the following groovy sample code:

```
def conn = adf.webServices.collaborations;
try
{
 def queryParams = ['q':'ObjectId='+OptyId]
 conn.dynamicQueryParams = queryParams
 def collabResponse = conn.GET()
 def weburl = ""
 if (collabResponse.count > 0 ) {
 weburl = collabResponse.items[0].Weburl;
 }
 return weburl;
}
 catch(Exception ex)
{
 println(conn.statusCode+""); //for diagnostic logging
 println(conn.httpErrorResponse+""); //for diagnostic logging
 throw ex;
 }
```

> **Note:** You can copy the scripts provided to set up the integration for custom objects, but you must replace the **OptyId** value with the ID field for the custom object in all the sample scripts.

8. Click **Validate**.
9. Click **Save and Close**.

Configure Smart Action Functions

Salespeople can define a smart action to take them directly to Microsoft Teams by entering a smart action in the Action Bar from their Opportunity page. This action will only be applicable if a team already exists for the selected opportunity record.

Salespeople can choose to create a team in Microsoft Teams by entering a smart action in the Action Bar from their Opportunity page in their sales application.

> **Note:** While the example refers to the opportunity object, the following set up steps are similar for other objects such as custom object as well.

As an administrator, you must define the Create Team in Microsoft Teams and Go to Microsoft Teams smart actions in Application Composer.

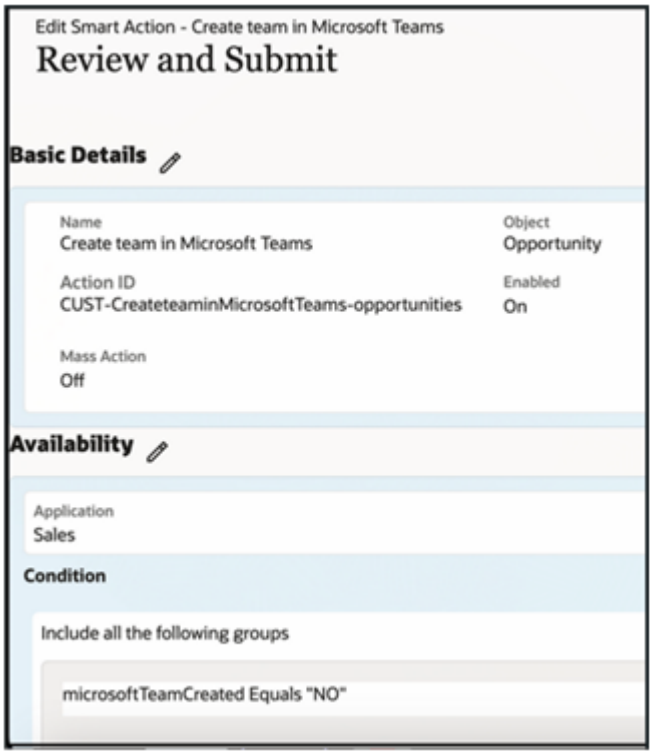
Define the Create Team in Microsoft Teams Smart Action

Here's how to define the Create Team in Microsoft Teams smart action function in Application Composer.

1. Navigate to **Configuration** > **Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. Click **Smart Actions**, available from **Common Setup**.
4. From the **Kind of Actions** page, choose **REST-based action**, and then click **Continue**.

5. Create the smart action with the following **Basic Details**:
   - **Name**: Create Team in Microsoft Teams
   - **Object**: Opportunity
   - **Enabled**: On
   - **Mass Action**: Off
   - For **Availability**, enter the following: **Application**: Sales
   - For **Condition**, enter `microsoftTeamCreated equals "NO"` in the **Include all the following groups** field.

   Here's an example of what the page looks like for the Opportunity object.



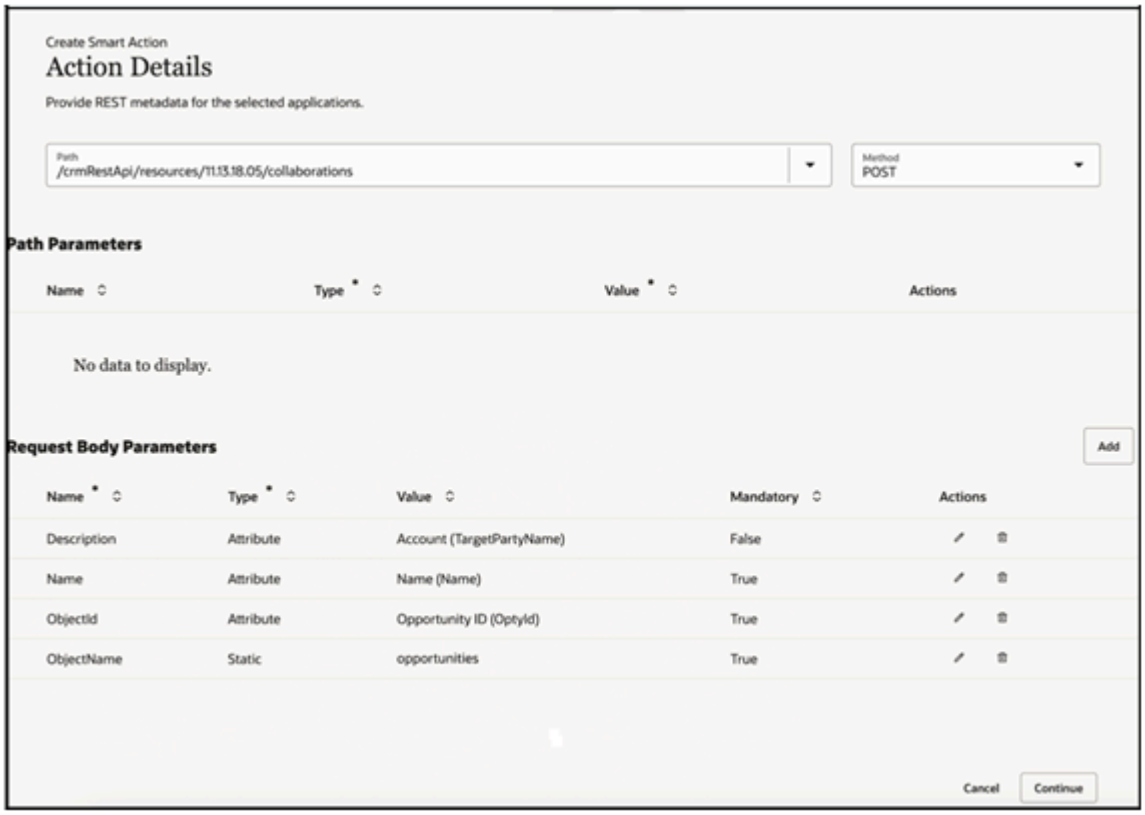
6. On the **Action Details** page, enter the REST metadata in the **Path** field of the selected smart action. `/crmRestApi/resources/11.13.18.05/collaborations .`
7. From the **Method** drop-down list, select **POST**.
8. Click **Add** and enter the following Request Body Parameters for the **Create Team in Microsoft Teams** smart action:

Object	Name	Type	Value	Required
Opportunity	ObjectName	Static	Opportunities	True
	ObjectId	Attribute	Opportunity ID (OptyId	True
	Name	Attribute	Name(Name)	True
	Description	Attribute	Account(TargetPartyName)	False

Object	Name	Type	Value	Required
	TeamRelationshipName	Static	To get the value, follow these steps: a. In a sandbox, go to **Application Composer**. b. Click **Relationships**. c. Copy the name of the relationship for the **Source Object** = Object of interest and **Target Object** = Resource The TeamRelationshipName parameter value is used to retrieve the team members on the record. It's only required if you're using a custom relationship for the standard objects or if you're setting up this integration for a custom object.	False
Custom	ObjectName	Static	Use API name of custom object. To get the value, follow these steps: a. In a sandbox, go to **Application Composer**. b. Expand node, then Custom Objects node. **Objects** node, then **Custom Objects** node. c. Click the custom object you want. d. On the **Overview** page for the Custom Object, copy the **API Name** value to use as the ObjectName	True
	ObjectId	Attribute	Custom object ID (ID field for the object)	True
	Name	Attribute	Name(Name)	True
	Description	Attribute	Account(TargetPartyName)	False
	TeamRelationshipName	Static	To get the value, follow these steps: a. In a sandbox, go to **Application Composer**. b. Click **Relationships**. c. Copy the name of the relationship for the **Source Object** = Object of interest and **Target Object** = Resource	False

Object	Name	Type	Value	Required

Here's a sample screenshot of what the Create Smart Action UI looks like:



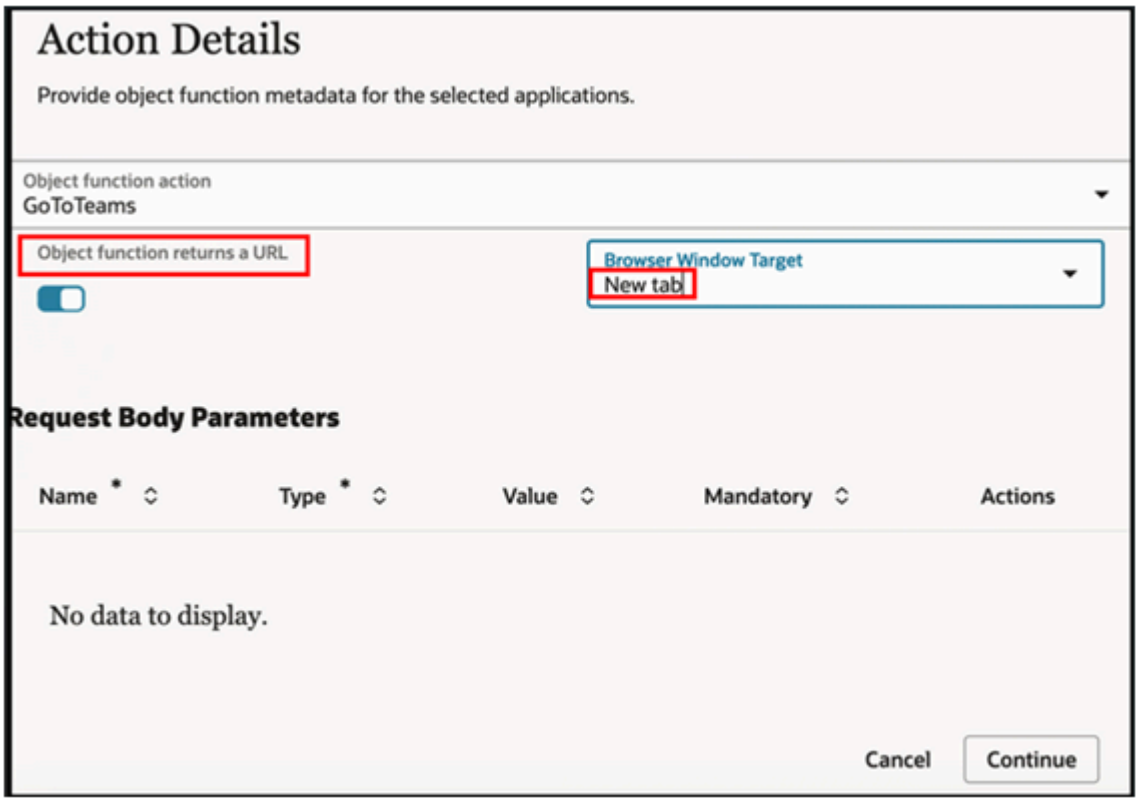
9. On the **Confirmation Message** page, you can define primary, secondary, and confirmation messages. Enter the following:
    - **Primary Message**: Your team will be created in Microsoft Teams in a few seconds.
    - **Secondary Message**: You can continue to use the application while it's being created. Please don't submit the action again.
    - **Continue Button Label**: OK
    - **Cancel Button Label**: Cancel
    - **Confirmation Message**: Your team OPPTY-{ATTRIBUTE_VALUE_NAME} in Microsoft Teams is created. Use the Go to Microsoft Teams action to go to the team.
10. Check the details on the **Review and Submit** page.
11. Click **Submit**.

Define the Go to Microsoft Teams Smart Action

Here's how to define the Go to Microsoft Teams smart action function in Application Composer.

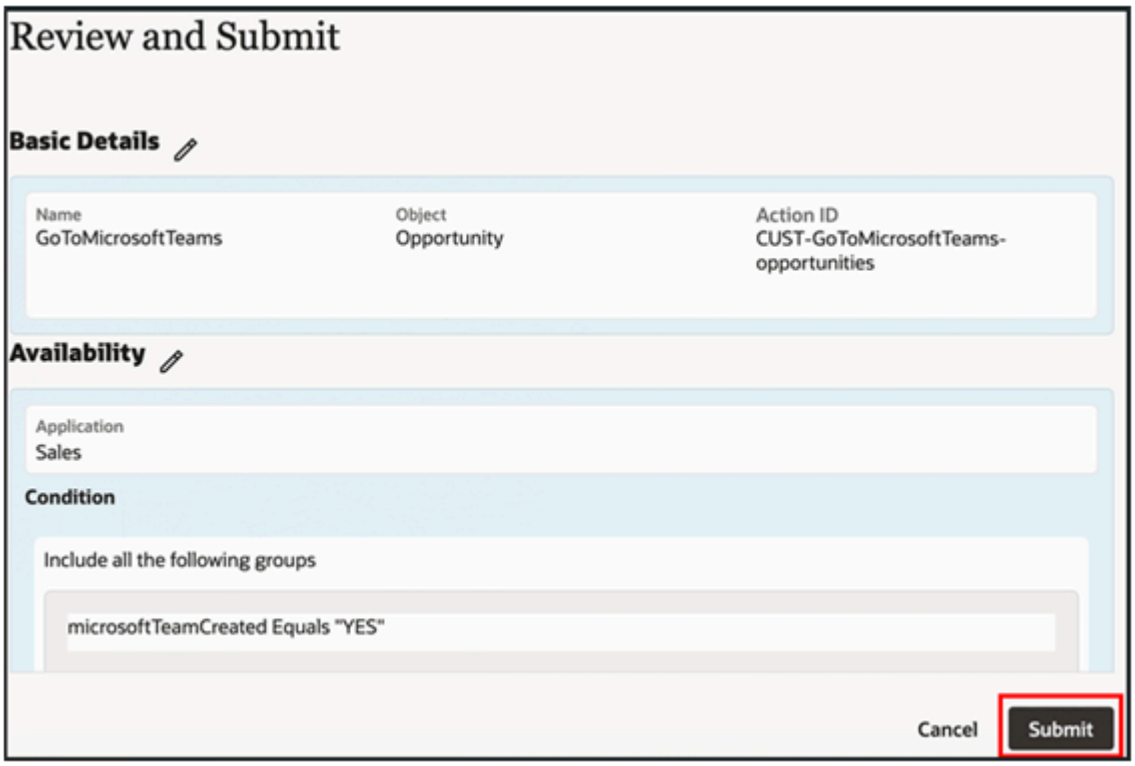
1. Navigate to **Configuration** > **Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. Click **Smart Actions**, available from **Common Setup**.

4. Create the smart action with the following **Basic Details**:
    - **Name**: Go To Microsoft Teams
    - **Object**: Opportunity
    - For **Availability**, enter the following: **Application**: Sales
    - For **Condition**, enter `microsoftTeamCreated equals "YES"` in the **Include all the following groups** field.
5. On the **Action Details** page, for **Object function action** field, select **GoToTeams** which you previously defined in the Create an Object Function Server Script topic.
6. Select the **Object function returns a URL** toggle.
7. In the **Browser Window Target** field, select **New Tab** from the drop-down list as shown in this sample screenshot of what the Create Smart Action UI looks like.



8. Click **Continue**.

9. Check the details on the **Review and Submit** page as shown:



10. Click **Submit**.

Salespeople can now add all existing members of their opportunity to a team in Microsoft Teams and start the collaboration directly from the Opportunities UI. This allows users of Microsoft Teams to interact with opportunity team members from their Sales application and add any required external resources with expertise to help win the deal. Using the **Create Team in Microsoft Teams** action in the opportunity, a dedicated opportunity team is created in Microsoft Teams for the opportunity team members. Once the team is created, users can use the **Go to Microsoft Teams** action to begin collaboration with the team.

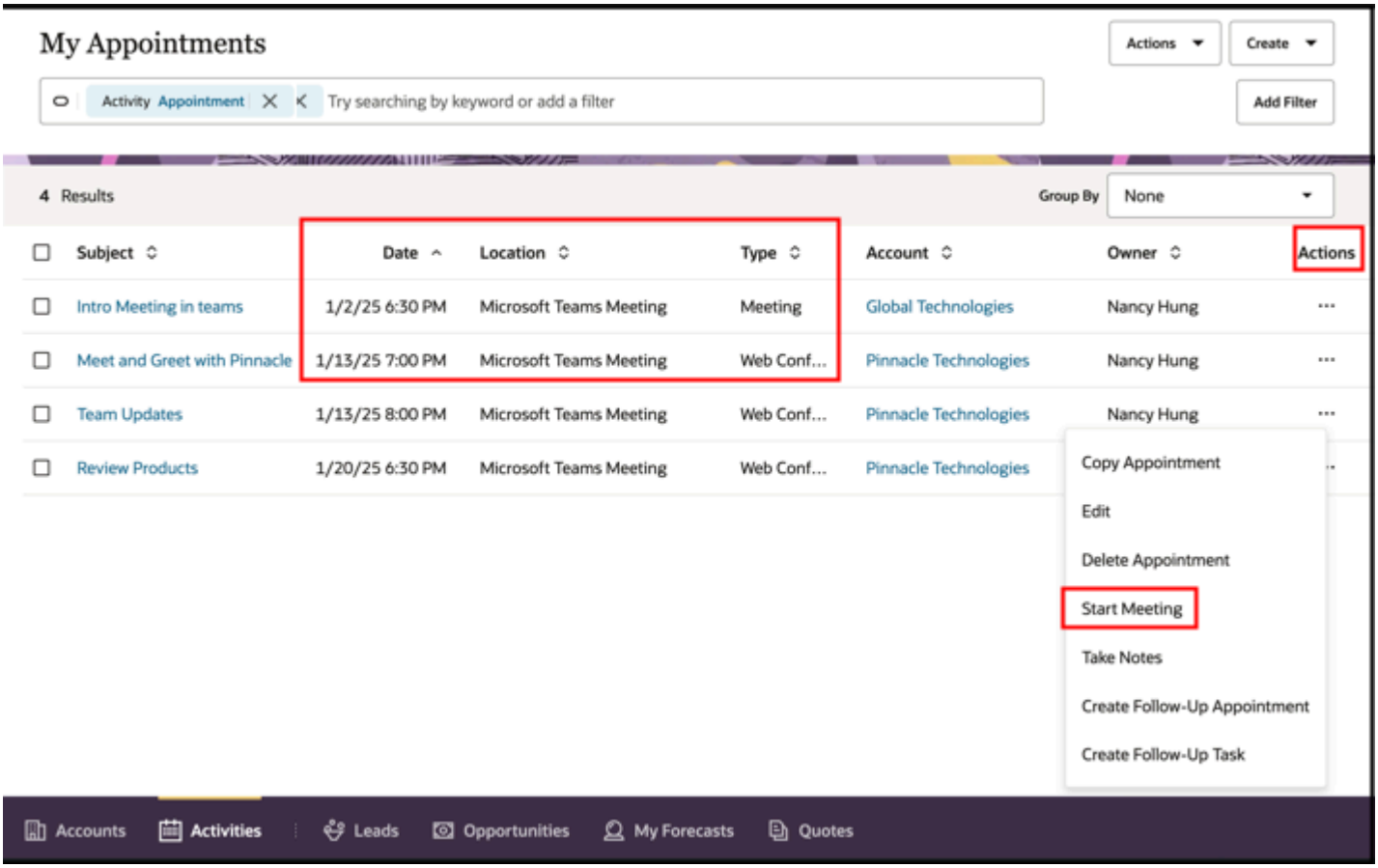
4 Microsoft Teams Integration for Web Conferencing

About Microsoft Teams Web Conference Integration

The Microsoft Teams web conference integration provides seamless connectivity between Microsoft Teams and the Oracle Sales application. With this integration, salespeople can:

- Start web conferencing meetings from within the Oracle sales application

  Here's a sample screenshot of the **Actions** menu from where salespeople can start web conference meetings.



- Verify and update meeting participants as contacts while the meeting is ongoing
- Take notes and record the meeting outcome
- View transcripts and recordings in the sales application
- Schedule Microsoft Teams meetings directly from Oracle Sales (available from 25C)

Watch this video *Integrate Microsoft Teams Web Conferencing with Oracle Sales* to get an overview of how you can launch Microsoft Teams web conferencing meetings in Oracle Sales and how you can auto capture interactions, recordings, and transcripts.

Overview of the Web Conference Integration Setup

Here's an overview of the basic setup that you must complete to enable, integrate, and verify a web conference app for your sales application for Microsoft Teams web conference integration users.

Step	Description	Where to get more information
1	Set up an application in Microsoft Azure with the application permissions required for this integration.	See *Set Up Microsoft Teams Web Conference Integration App in Microsoft Azure*.
2	Configure an application access policy in Microsoft.	See *Create an Application Access Policy for Microsoft Teams Web Conference Integration*.
3	Use **Manage Web Conference Integration** to configure the settings for Microsoft Teams web conference integration.	See *Configure Web Conference Integration Settings in the Sales Application* .
4	Enable the web conference activity type. This setup is required to allow salespeople to open Microsoft Teams web conference with the click of the **Start Web Conference** button.	See *Set Up Web Conference Activity Type*.
5	Run **Synchronize Web Conference Data** process to sync past web conference meeting details, such as meeting participants, transcripts and recordings into your Sales application from Microsoft 365.	See *Schedule the Web Conference Process*.

Set Up Microsoft Teams Web Conference Integration App in Microsoft Azure

To create the Microsoft Teams web conference integration, you must first create the Microsoft Azure application.

Follow these steps:

1. As an Azure or Exchange administrator, sign in to the Microsoft Azure portal at:

   `https://portal.azure.com`

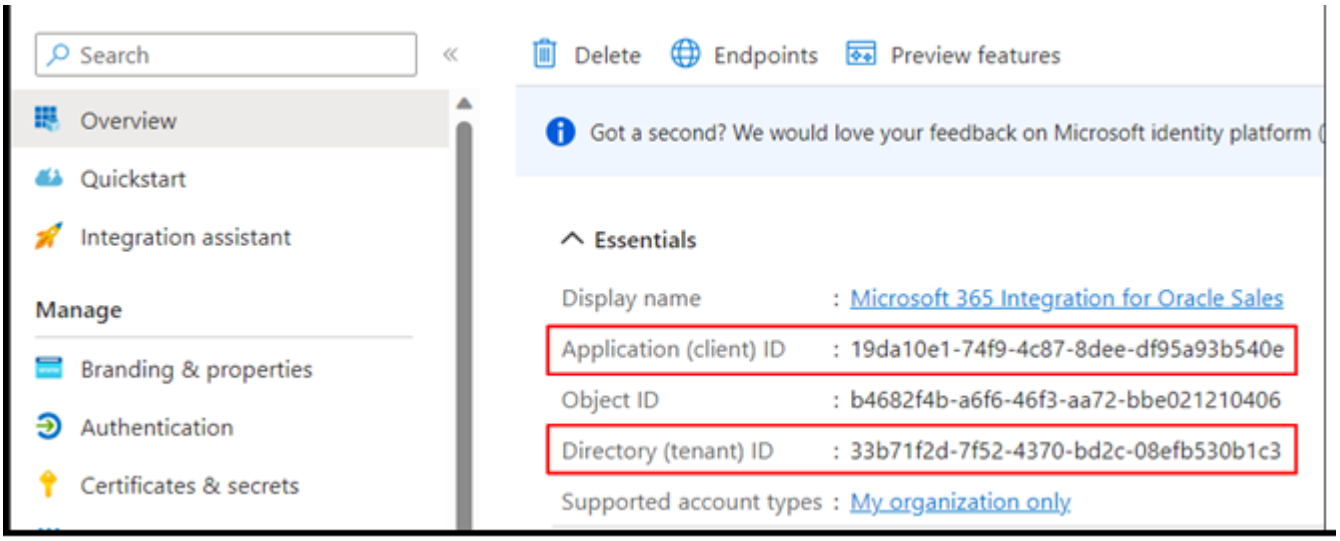
   > **Note:** If you're a first-time user, then you must sign up for your Microsoft Entra ID (formerly Azure Active Directory) account.

2. On the home page of Microsoft Azure, search for

   `App registrations`

3. Click **App registrations**. The Applications list page appears.
    a. Click **+ New registration**.
    b. For **Name**, enter a name for the application, for example, `Microsoft Teams Web Conference Integration`.
    c. From the **Supported account types** area, select **Accounts in this organizational directory only (single tenant)**.
    d. Leave **Redirect URI (optional)** blank.
    e. Click **Register** to create the Microsoft Azure application.

    When completed, the application appears on the App registrations page.
4. The Application details page displays the Application (client) ID and Directory (tenant) ID similar to the following sample screenshot:

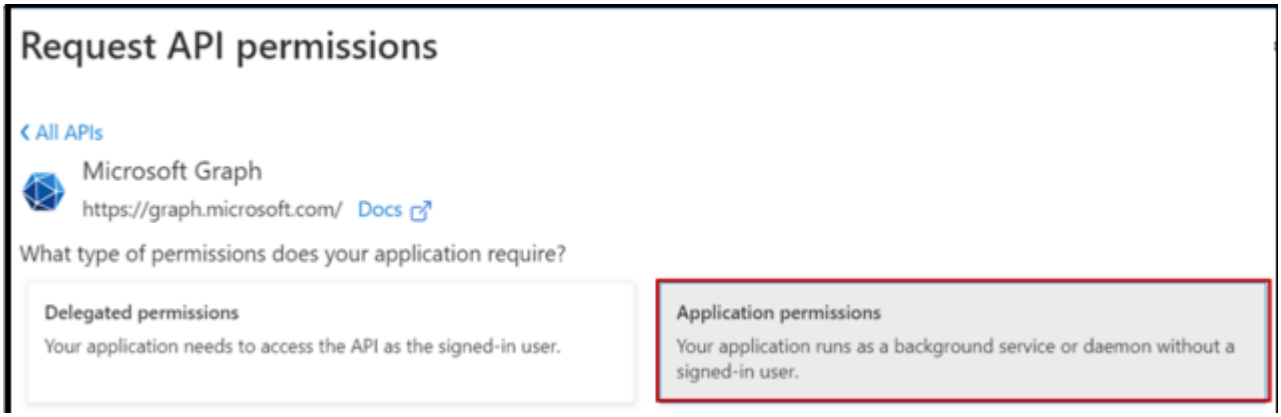


> **Note:** Keep a note of these values because you insert these IDs later on in the Manage Web Conference Integration task page. See *Configure Web Conference Integration Settings in the Sales Application* section for more information.

5. Next, click **Certificates & secrets**, and from the **Add a client secret** section, click **+ New client secret**.
    - Enter a name, for example, Web Conference Integration Secret.
    - Select a duration for the expiration of the Client Secret. Microsoft currently allows a maximum validity period of 24 months (730 days) from the start date. Oracle recommends selecting the maximum tenure of **24 months** to minimize disruptions.

      A new Client Secret must be generated before the expiration date and saved to Oracle Sales to ensure the integration continues to work. A new client secret will need to be generated before the expiration date and saved to Oracle so that the integration can continue to work. If you select **Custom**, enter the start and end dates, which must be within 24 months.
6. Click **Add**.

   The application generates a **Client Secret Value**. Keep a note of this value because you need to insert this in the Manage Web Conference Integration task page later on.
7. Click **API permissions** and click **Add a permission**. The **Request API permissions** pane appears.
8. Scroll to the end of the pane and click **Microsoft Graph**.
9. Click **Application permissions** as shown in this screenshot:



Select the following permissions:

- `Calendars.ReadWrite`
- `User.Read.All`
- `Chat.Read.All`
- `OnlineMeetingArtifact.Read.All`
- `OnlineMeetings.ReadWrite.All`
- `OnlineMeetingTranscript.Read.All`

10. Click **Add permissions**.
11. Click **Grant admin consent for <name>** and then click **Yes** on the confirmation dialog box.

    The Status of the permissions changes to **Granted**.

Create an Application Access Policy for Microsoft Teams Web Conference Integration

You must set up the application access policy for the application you created for this integration, and then grant the policy to each of the Microsoft users who need access to this integration.

Here's an overview of the setup steps:

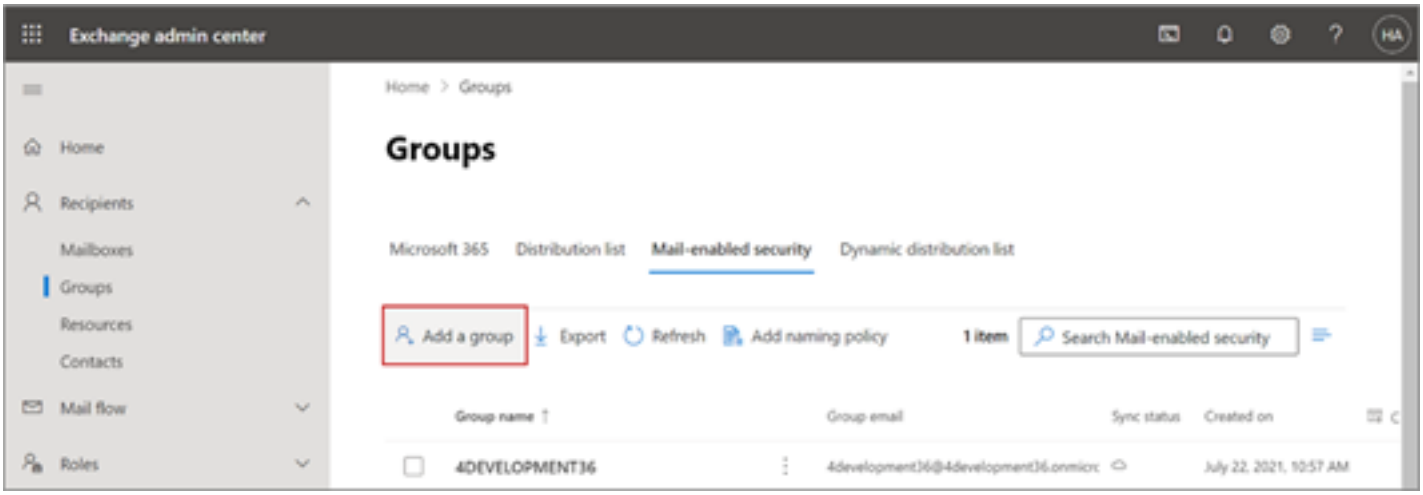
1. Create a mail-enabled security group in the Microsoft Exchange admin center.
2. Create an application access policy in **Microsoft PowerShell** for the security group you created.

Create a Mail-Enabled Security Group

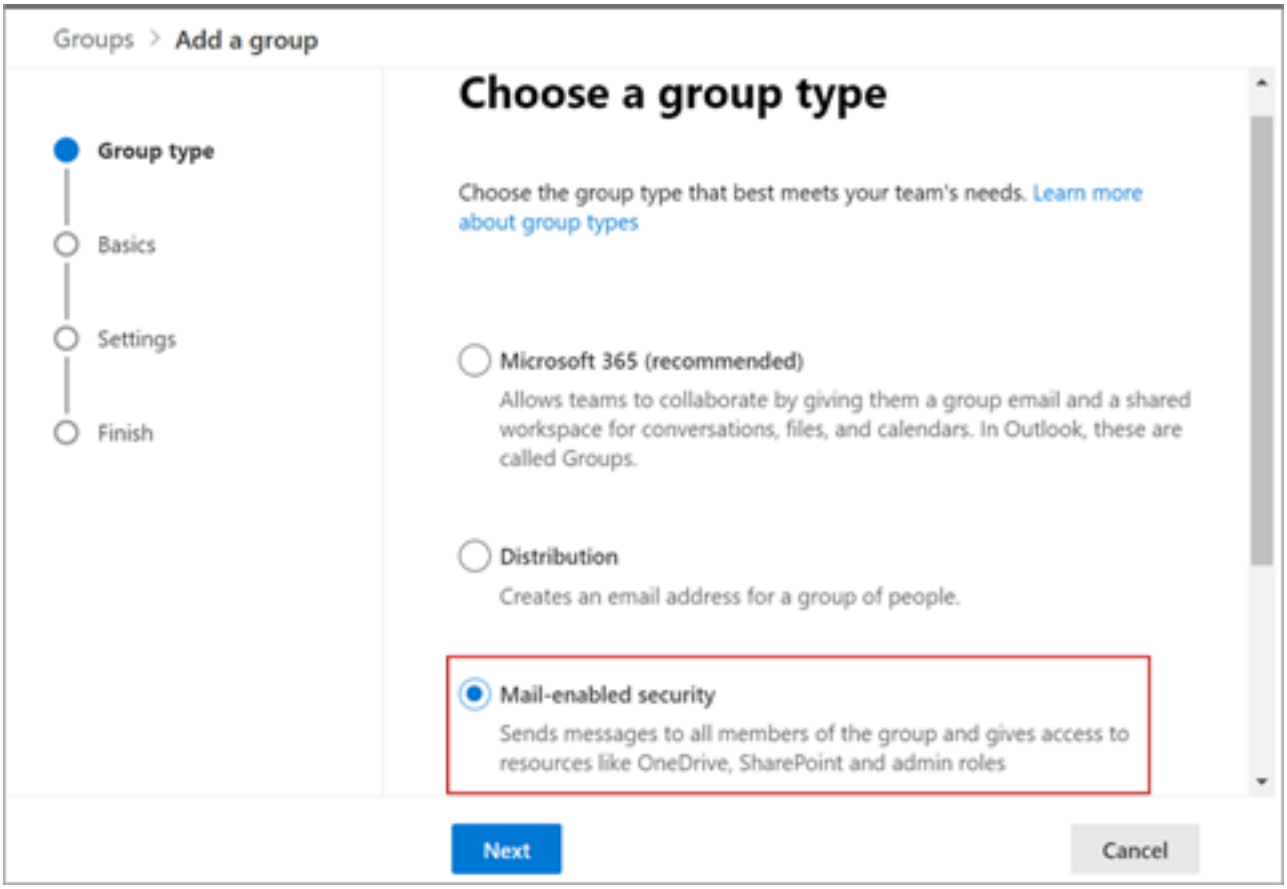
Here are the steps to set up a mail-enabled security group.

1. Sign in to the Exchange Admin Center at `https://admin.exchange.microsoft.com/#/`.
2. Navigate to **Groups** > **Mail-enabled security**.

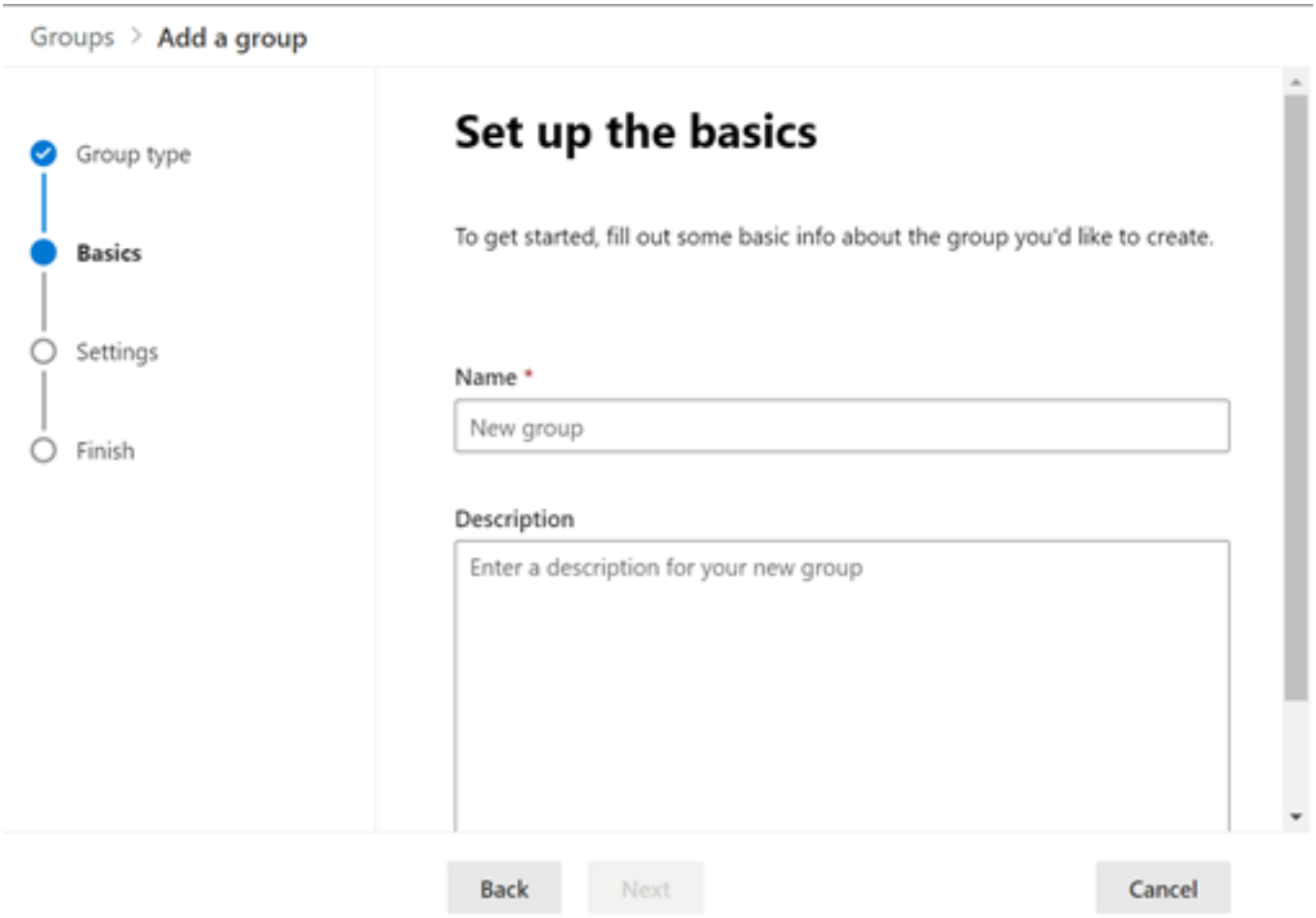
3. Click **Add a group** as shown.



4. In the **Group type** section, select **Mail-enabled security** and click **Next**, as shown:

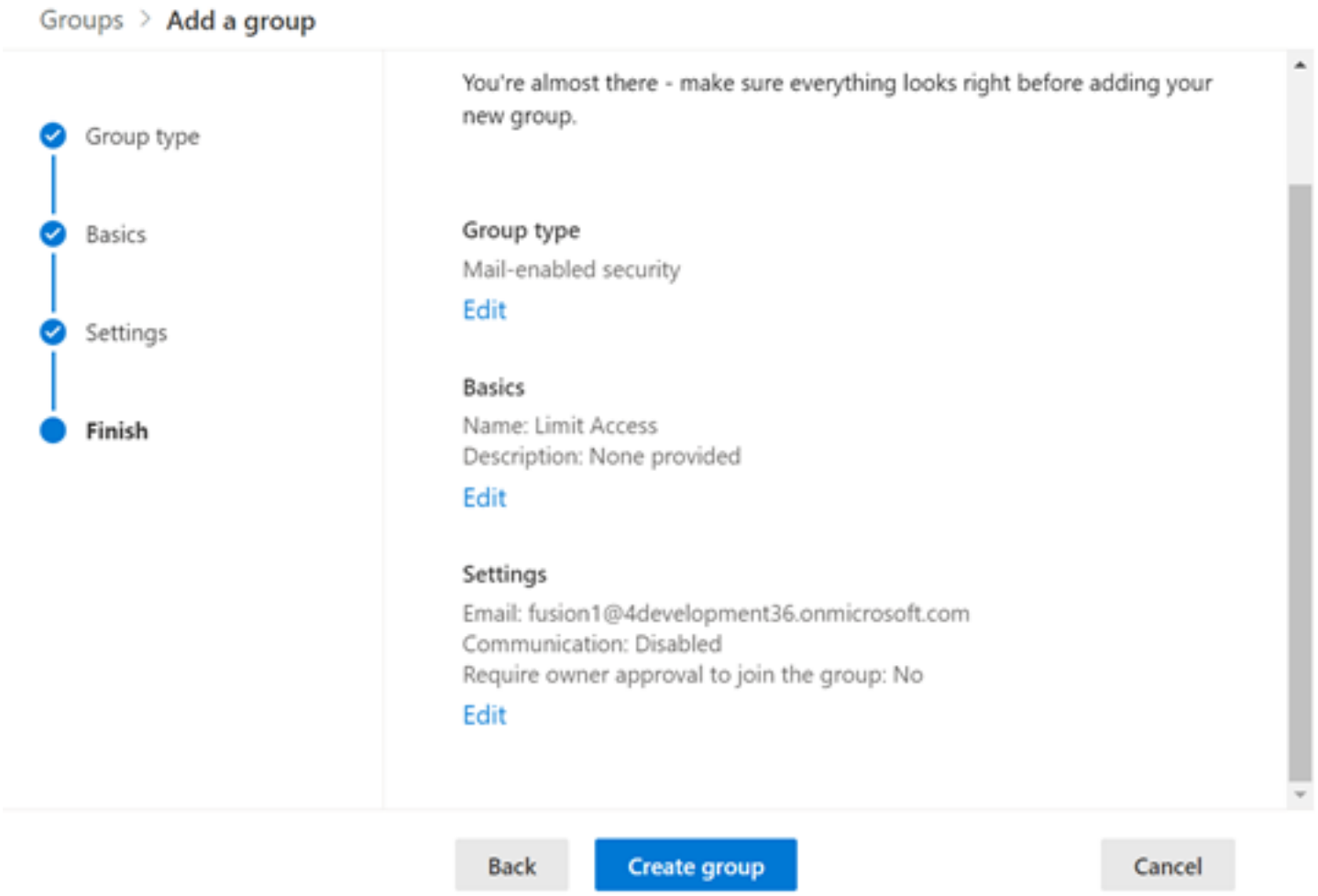


5. In the **Set up the basics** section, enter a name and description, and click **Next** as shown:

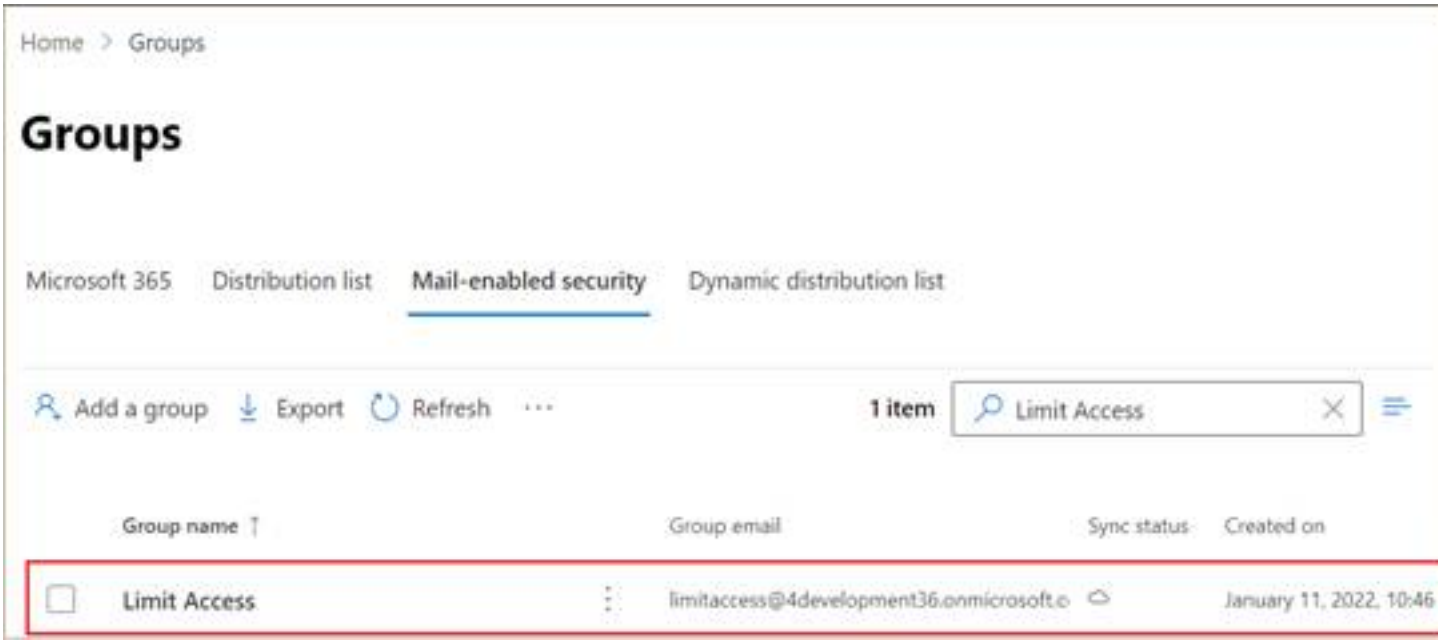


6. In the **Settings** section, enter the group email address, and configure the following:
   a. **Communication**: Don't select the checkbox.
   b. **Approval**: Select the **Require owner approval to join the group** checkbox.

7. In the **Finish** section, verify all the details and click **Create group** as shown:

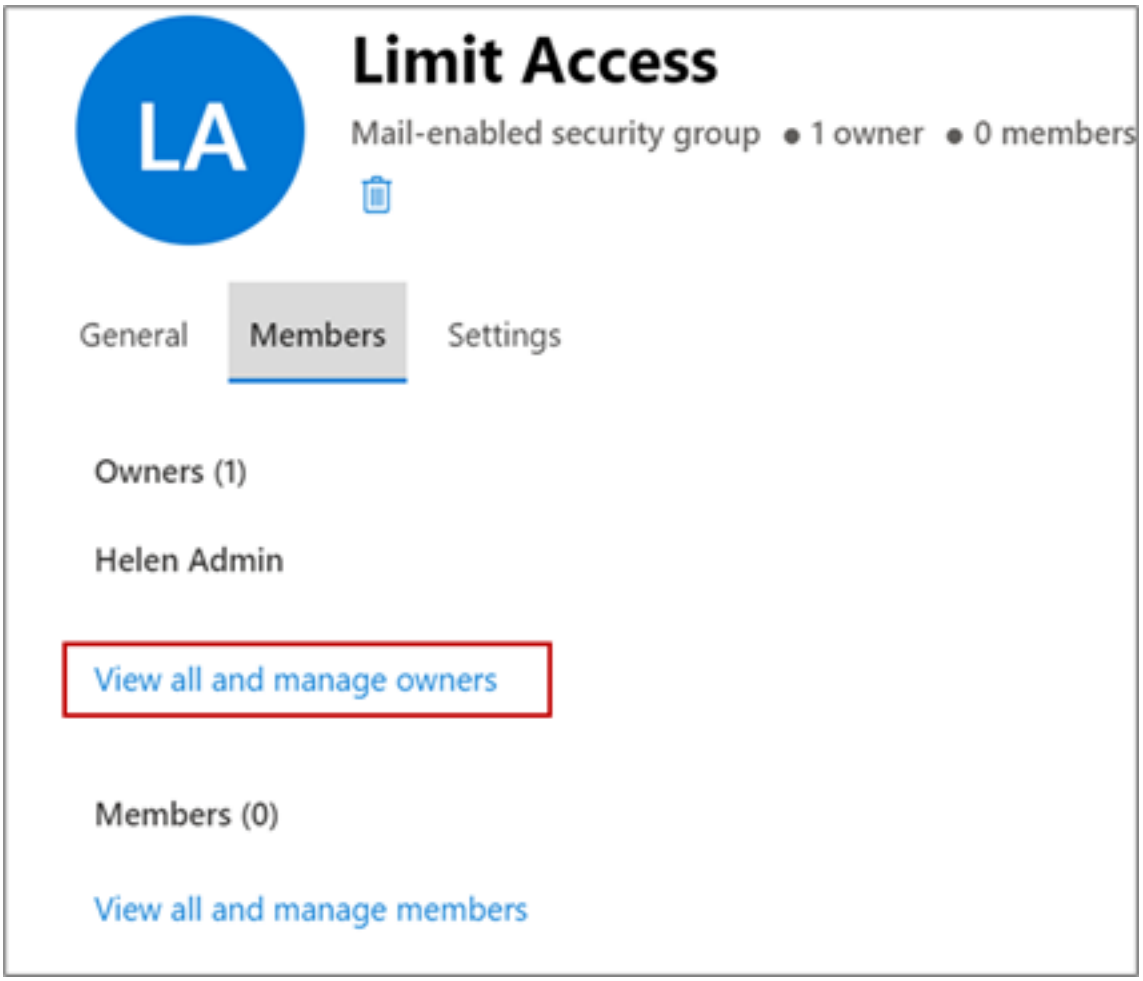


8. The mail-enabled security group you created shows up in the Groups list as shown:

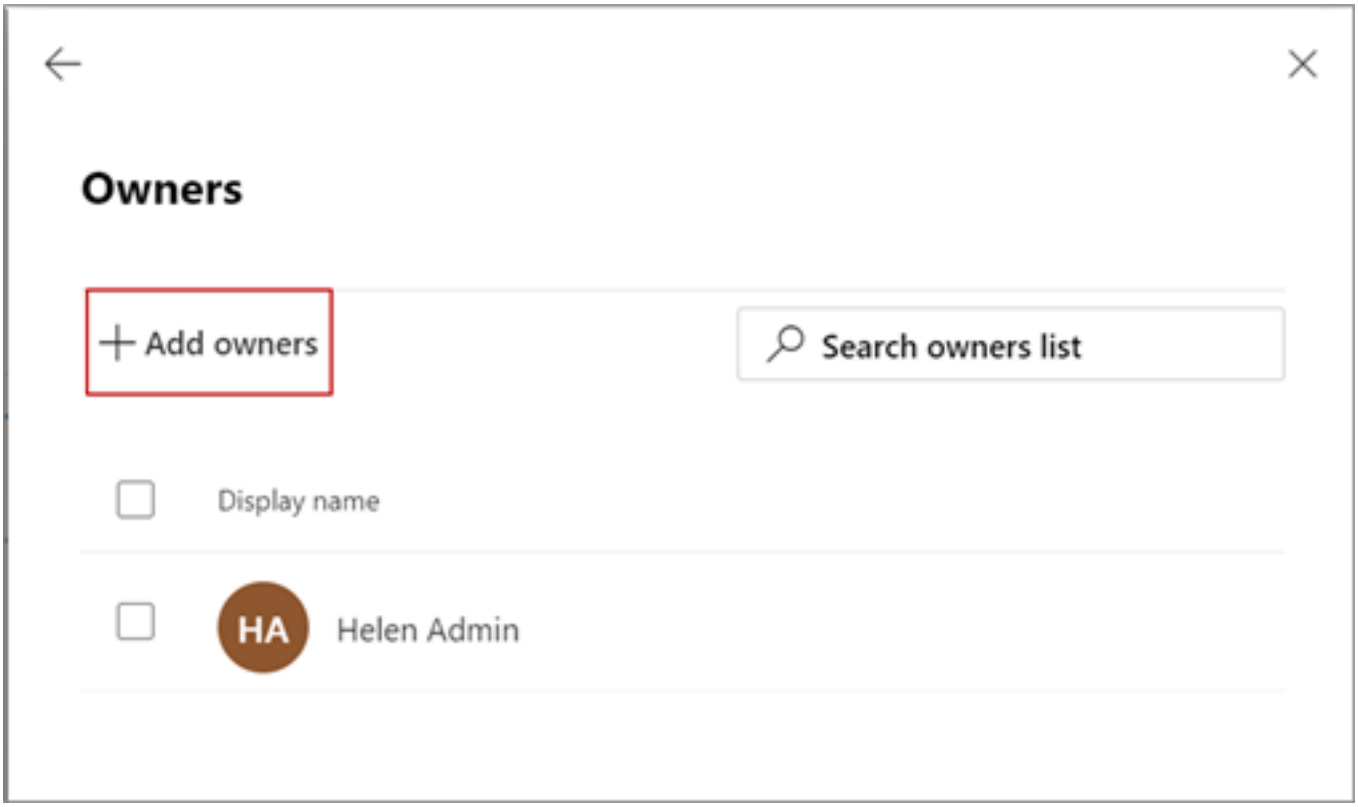


9. Next, you need to assign owners and add members to the group. To do that, click the group name.

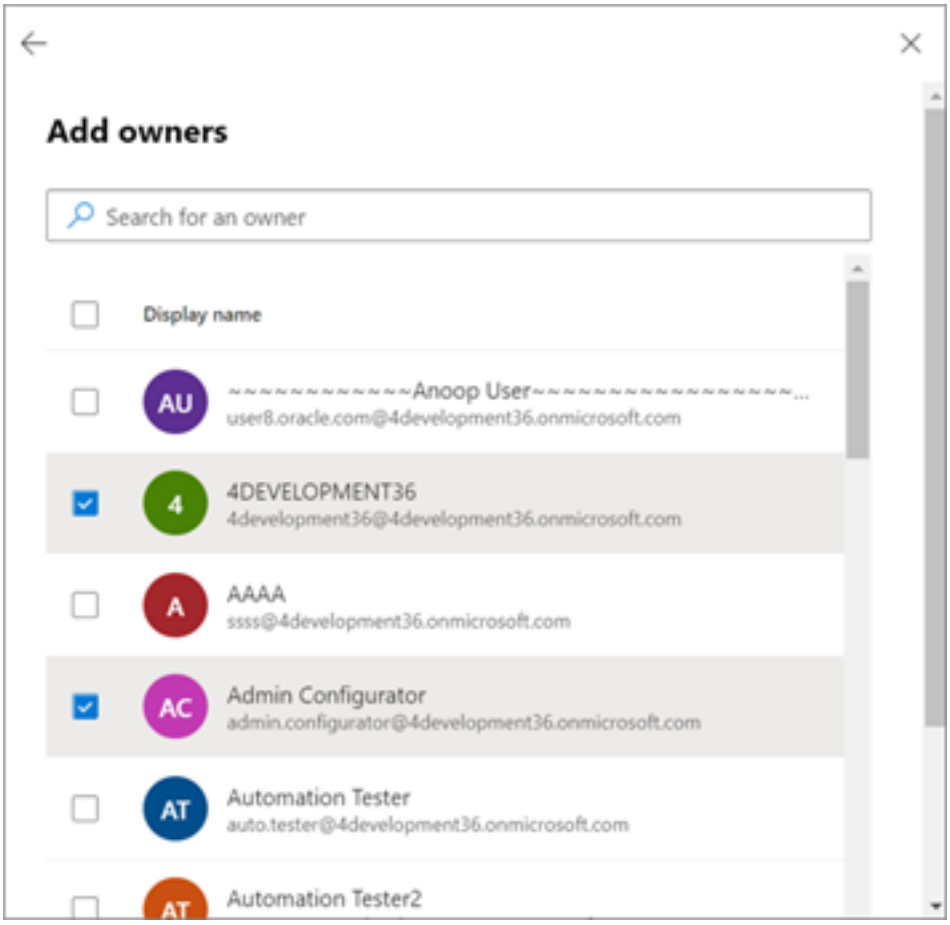
10. On the Group Details window, click the **Members** tab.
    a. You're the owner of the group you created. To add more owners, click **View all and manage owners**. For example, the following image shows the Limit Access window with the Members tab highlighted, along with the View and manage all owners link.



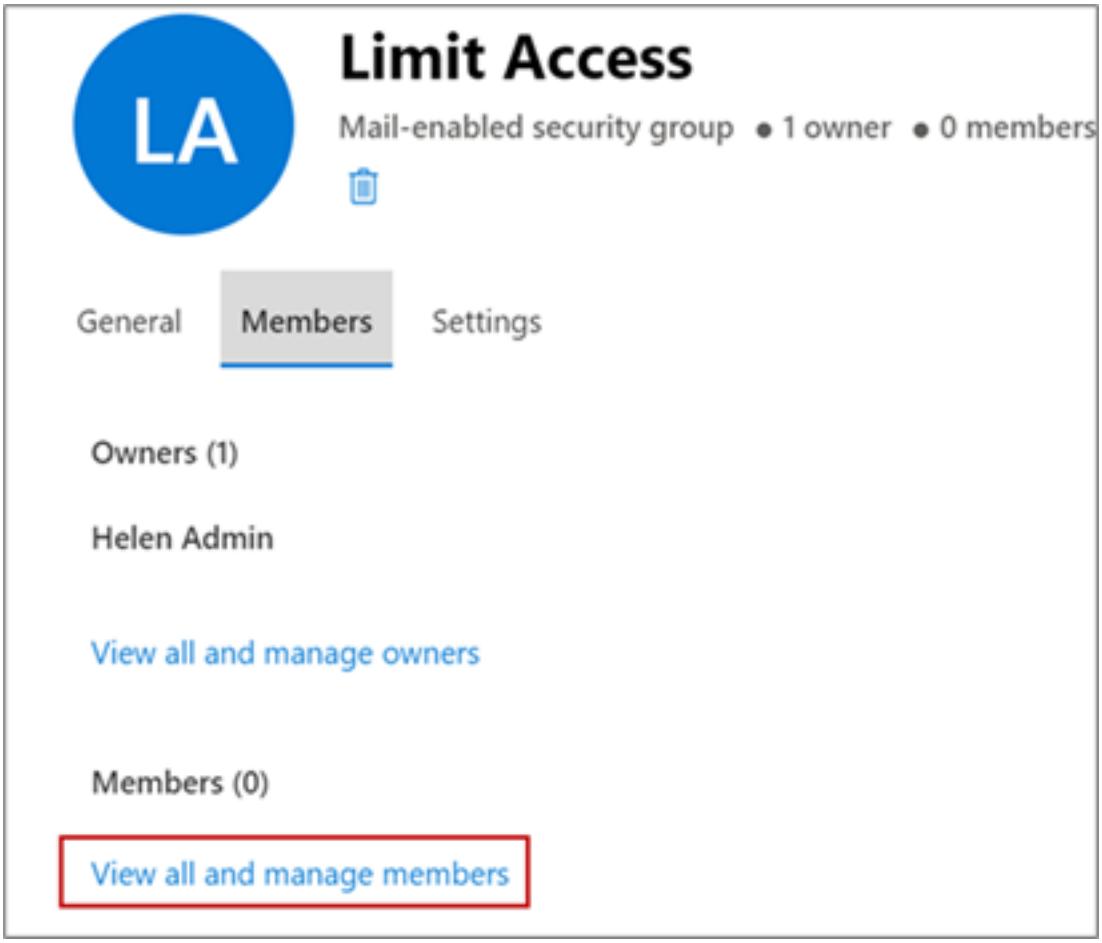
    b. On the Owners page, click **Add owners** as shown:



c. Select the group owners from the list and click **Add** as shown:



d. Click the back arrow to go back to the Members tab and click **View all and manage members** to add users you want to give seamless access to as shown:



e. On the Members page, click **Add members**.
f. Select the members from the list and click **Add**.
g. Click the back arrow to go back to the Members tab

11. Close the group.

The mail-enabled security group is ready for use.

Create an Application Access Policy

Follow these steps:

1. Open **Windows PowerShell**.
2. Run the following commands in **Windows PowerShell**.
   - `Install-Module -Name MicrosoftTeams`
   - `Connect-MicrosoftTeams`

     If you haven't authenticated yet, **Windows PowerShell** will prompt you to sign in with your Microsoft Azure Administrator credentials.

3. Once authenticated, go back to **Windows PowerShell** and create an application access policy by running the following command and replacing **AppId**:

```
New-CsApplicationAccessPolicy -Identity XXXXXXXXXXX -AppIds "XXXXXXXXXXX" -Description "XXXXXXXXXXX"
```

`AppID` represents the application (client) ID of the registered Azure App for this integration. `Identity` represents the name of the policy and `Description` is the optional policy description.

4. Once the policy is created, grant the policy to the Microsoft security group by running the following command and replacing the arguments for the **PolicyName** and **groupID**.

```
Grant-CsApplicationAccessPolicy -PolicyName XXXXXXXXXXX -Group "groupID"
```

`PolicyName` represents the name of the policy to be assigned and `groupID` specifies the group used for the group policy assignment.

> **Note:** To retrieve `groupID` of the Microsoft security group created for this integration, run the following command in **Windows PowerShell**:
>
> ```
> Get-MgGroup -Search
>
> "displayName:name" -ConsistencyLevel eventual
> ```
>
> Replace the attribute for `displayName` with the full or partial name of your mail-enabled security group (you can use a keyword to search).

It might take about 30 minutes for your changes to be reflected.

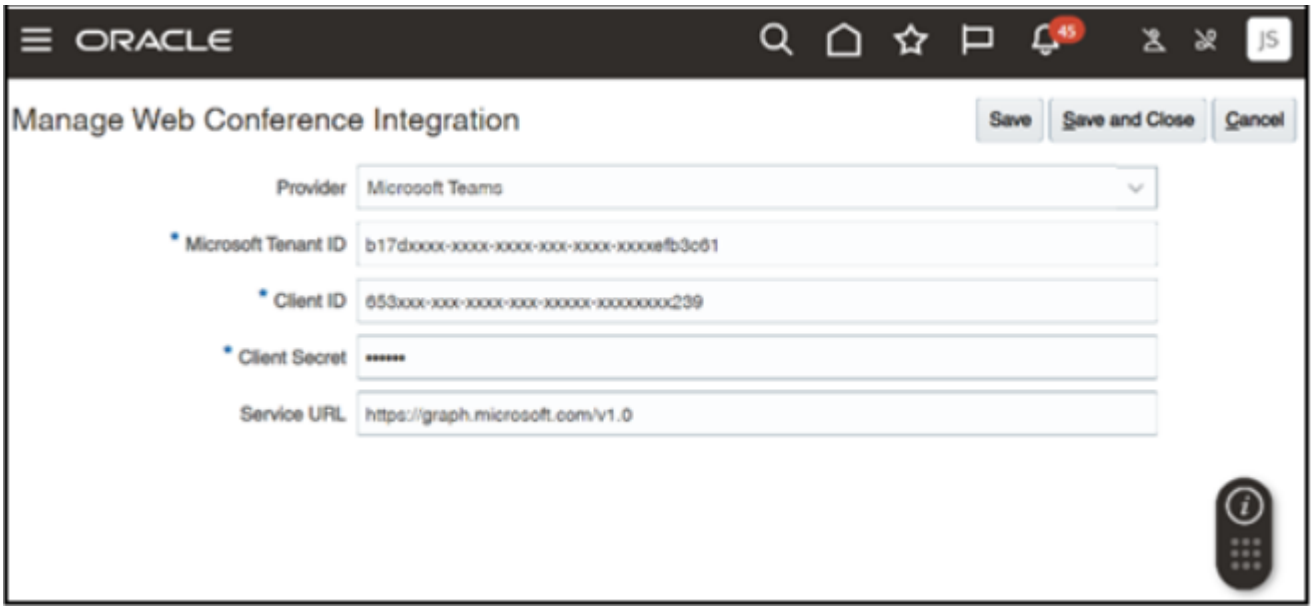
Configure Web Conference Integration Settings in the Sales Application

You're now ready to complete and activate the Microsoft Teams web conference integration for your sales application. Use the **Manage Web Conference Integration** task to configure the integration settings.

Follow these steps:

1. Sign in to the application as a setup user.
2. Click **Navigator** > **My Enterprise** > **Setup and Maintenance**.
3. In the Setup and Maintenance work area, go to click **Tasks** > **Search**.

4. Search for the **Manage Web Conference Integration** task and click the matching result. The following screen is displayed:



5. In the **Provider** field, select **Microsoft Teams** from the drop-down list.
6. For the **Microsoft Tenant ID**, **Client ID** and **Client Secret** fields, use the associated values from the App registered in the Azure setup. See steps 4 and 6 in the *Set Up Microsoft Teams Web Conference Integration App in Microsoft Azure* topic.
7. For the **Service URL** field, always use `https://graph.microsoft.com/v1.0`
8. Click **Save and Close**.

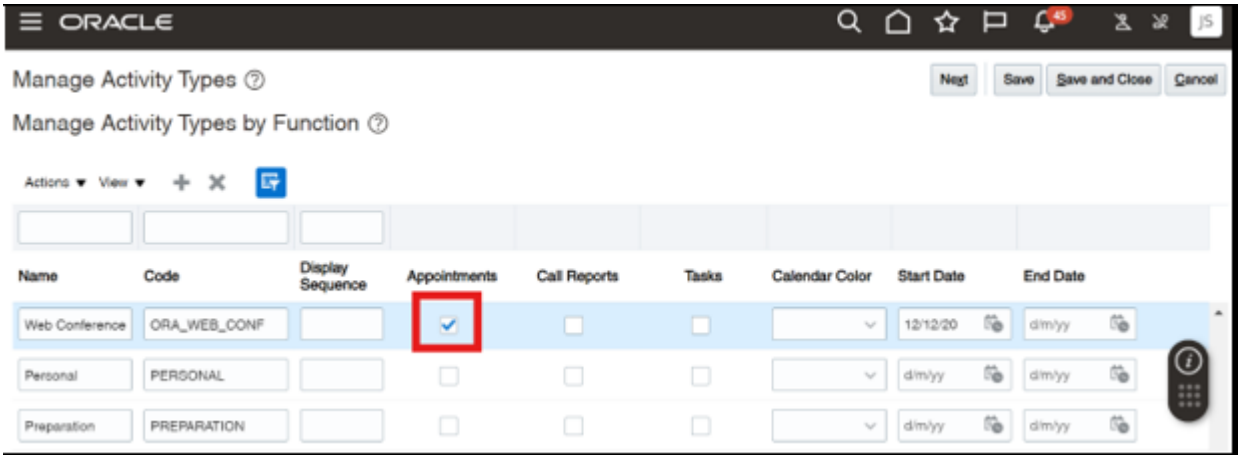
Set Up Web Conference Activity Type

Use the Manage Web Conference Integration task to configure the settings for the web conference activity type.

Follow these steps:

1. Navigate to **My Enterprise** > **Setup and Maintenance** .
2. In the Setup and Maintenance work area, go to click **Tasks** > **Search**.
3. Search for the **Manage Activity Type and Subtype** task and click the matching result.

4. Enable activity type **Web Conference** with **Code** `ORA_WEB_CONF` for **Appointments** highlighted as shown:



5. Click **Save and Close**.

Schedule the Web Conferencing Process

Run the **Synchronize Web Conference Data** scheduled process to sync past web conference meeting details such as meeting participants, transcripts and recordings into your Sales application from Microsoft 365.

Here's how to run and schedule the processes:

1. Sign in to the application as a setup user.
2. Navigate to **Tools** > **Scheduled Processes**.
3. Click **Schedule New Process**.
4. In the **Schedule New Process** dialog window, ensure **Job** (not Job Set) is selected.
5. Select the drop-down list arrow next to the search box and click **Search**.
6. In the Search and Select: Name dialog window, enter **Synchronize Web Conference Data** and click **Search**.
7. Click the name of the job in the search results and click **OK**.
8. Click **OK** again.
9. In the Process Details window, select the options you want. For example:
    a. Set the **Sync Look-Back Period in Minutes**.

       For example, if this time is set to 30 minutes, then each synchronization run will sync meetings that were started in the last 30 minutes.
    b. Select the **Include Transcripts and Recordings** checkbox to sync web conference transcripts and recordings to your sales application.
10. To run the job immediately, select your options and click **Submit**.
11. To set up a schedule for the job, click **Advanced** and click the **Schedule** tab. Then enter a schedule before submitting or saving changes.

    > **Note:** Oracle recommends scheduling this job to be run every 30 minutes.

12. Click **Submit**.