Informatica® Cloud Application Integration
September 2024

Create NetSuite Sales Orders from Dynamics 365 Opportunities

Informatica Cloud Application Integration **Create NetSuite Sales Orders from Dynamics 365 Opportunities**
September 2024

Publication Date: 2024-10-01

Table of Contents

Preface

Use *Create NetSuite Sales Orders from Dynamics 365 Opportunities* to learn how to create a NetSuite sales order from a Dynamics 365 opportunity based on a webhook request from Dynamics 365. This guide assumes that you have an understanding of the Dynamics 365 Connector, NetSuite Connector, and Email Connector concepts.

CHAPTER 1

Introduction to Create Netsuite Sales Orders from Dynamics 365 Opportunities recipe

The Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe is based on REST and SOAP APIs. Use the recipe to create a NetSuite sales order from a Dynamics 365 opportunity based on a webhook request from Dynamics 365 as an incoming parameter.

The process is triggered by a webhook to check the status of the Dynamics 365 opportunity. If the opportunity status is Won, the process retrieves and stores the key information, such as the associated account, currency, and products, and creates all the products from the opportunity in NetSuite. Otherwise, the process is terminated. If a product with the same name already exists, the process retrieves the product's ID instead of creating a new product.

The process then creates or updates the customer record in NetSuite based on the opportunity's account information. The process searches for an existing sales order in NetSuite with the Dynamics ID custom field field that match the opportunity ID. If the sales order exists, an email notification is sent. Otherwise, a new sales order is created based on the data from the Dynamics 365 opportunity.

The process then sends an email notification with the results to confirm the successful creation of the sales order or to report errors encountered during the process.

The process does not update the existing sales orders. If an existing sales order is found, a notification is sent through email, but no further action is taken.

Prerequisites

To create a NetSuite sales order from a Dynamics 365 opportunity, the following prerequisites must be met:

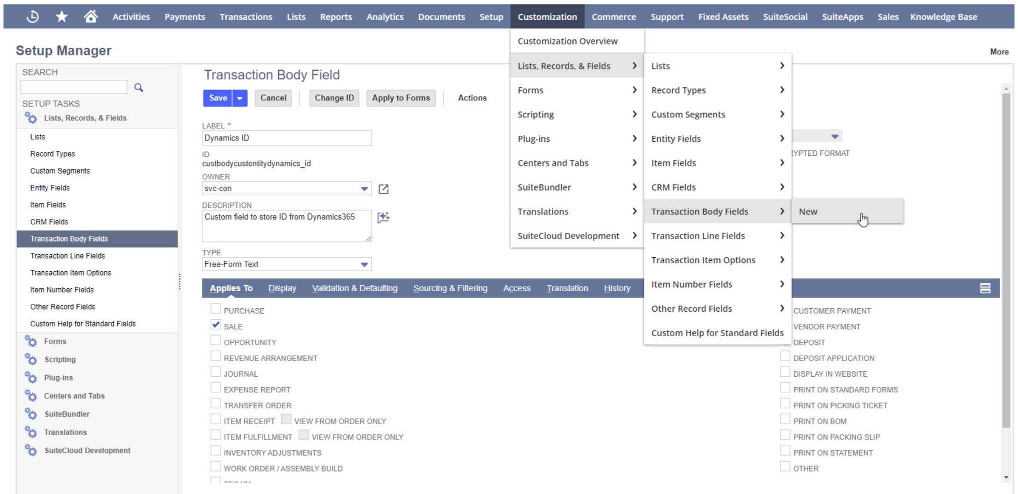
- Create a custom field in the NetSuite sales order entity
- Create a custom field in the Dynamics 365 account entity
- Create a webhook in Dynamics 365

Creating a custom field in the NetSuite sales order entity

Create a `Dynamics ID` custom field in the NetSuite sales order entity.

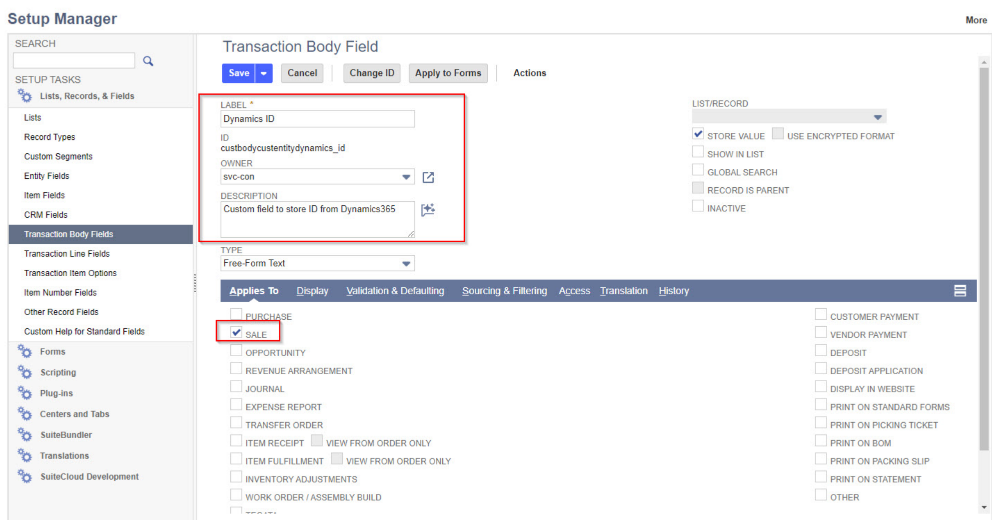
1. Log in to the NetSuite organization.

2. Go to **Customization** > **Lists, Records, & Fields** > **Transaction Body Fields** > **New**.
   The following image shows the **Transaction Body Field** page:



3. On the **Transaction Body Field** page, enter the value as `Dynamics ID` in the **LABEL** field and `custbodycustentitydynamics_id` in the **ID** field.

4. Select **SALE** in the **Applies To** tab as shown in the following image:



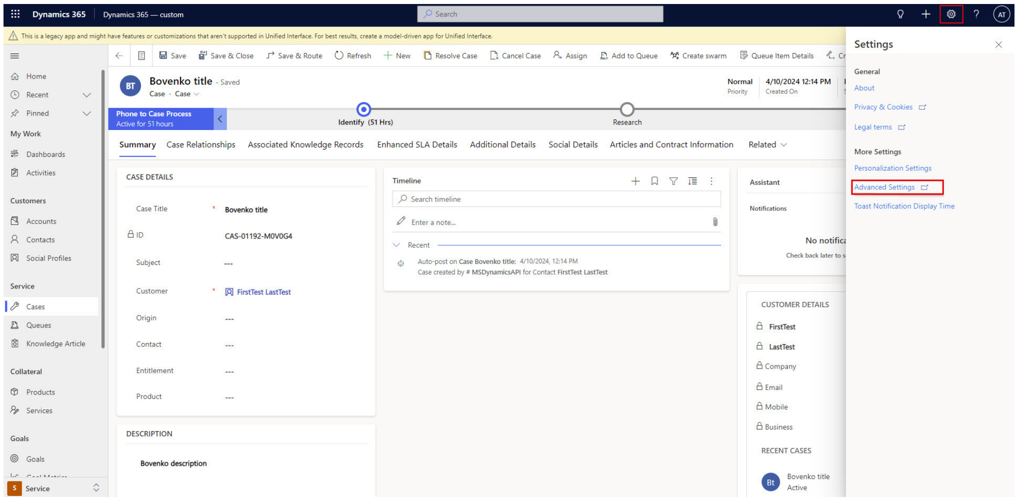
   These values are used for searching the matching sales order.

5. Click **Save**.

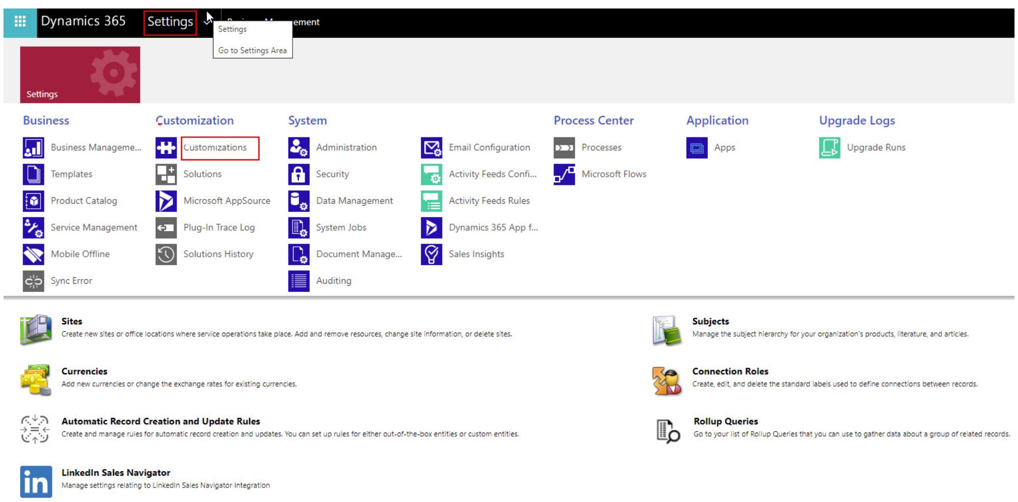
Creating a custom field in Dynamics 365 account entity

Create a `new_netsiteid` custom field in the Dynamics 365 account entity.

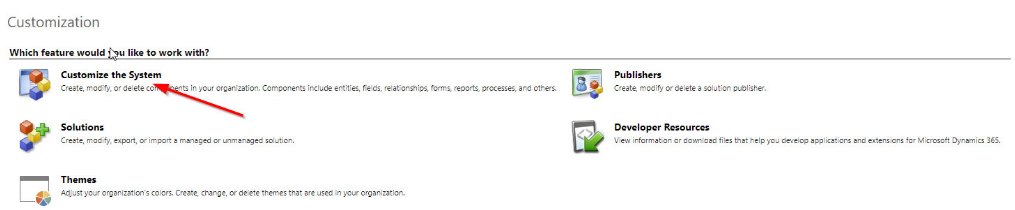
1. Log in to the Dynamics 365 organization.
2. Go to **Settings** > **Advanced Settings**.
   The following image shows the **Advanced Settings** option on the **Dynamics 365** page:



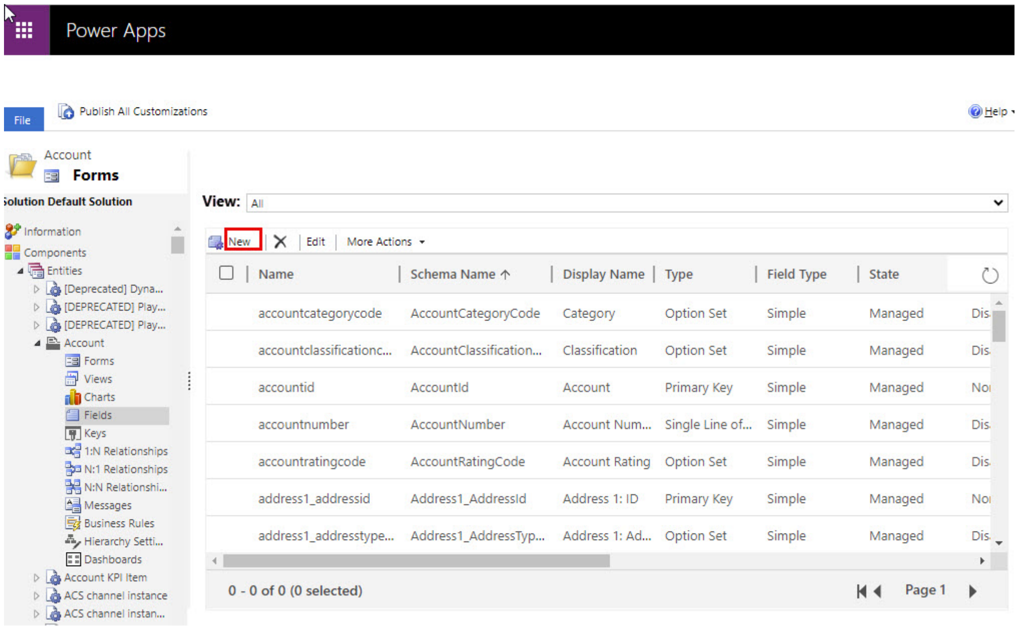
3. On the **Settings** page, click **Customizations**.
   The following image shows the **Customizations** option on the **Settings** page:



4. Click **Customize the System**.
   The following image shows the **Customize the System** option on the **Customization** page:

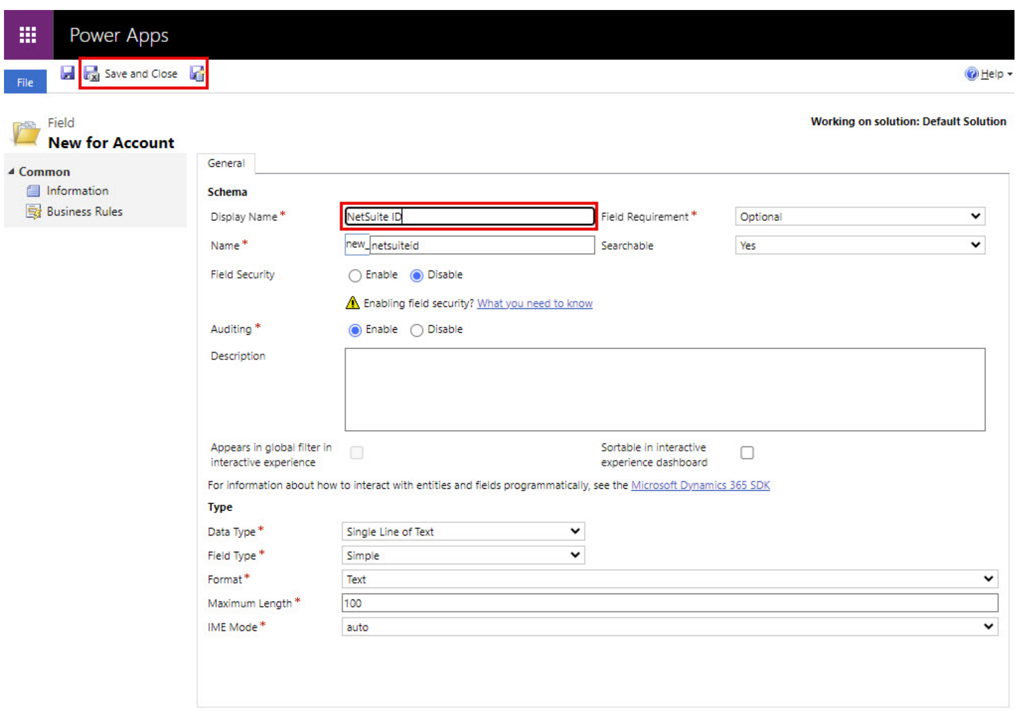


5. Go to **Components** > **Entities** > **Account** > **Fields**, and then click **New**.
   The following image shows the **New** button on the **Account Fields** page:



6. In the **Display Name** field, enter the value as **NetSuite ID** and click **Save and Close**. The value in the **Name** field is automatically set to **new_netsuiteid**.

   The following image shows the **New for Account** page:



Creating a webhook in Dynamics 365

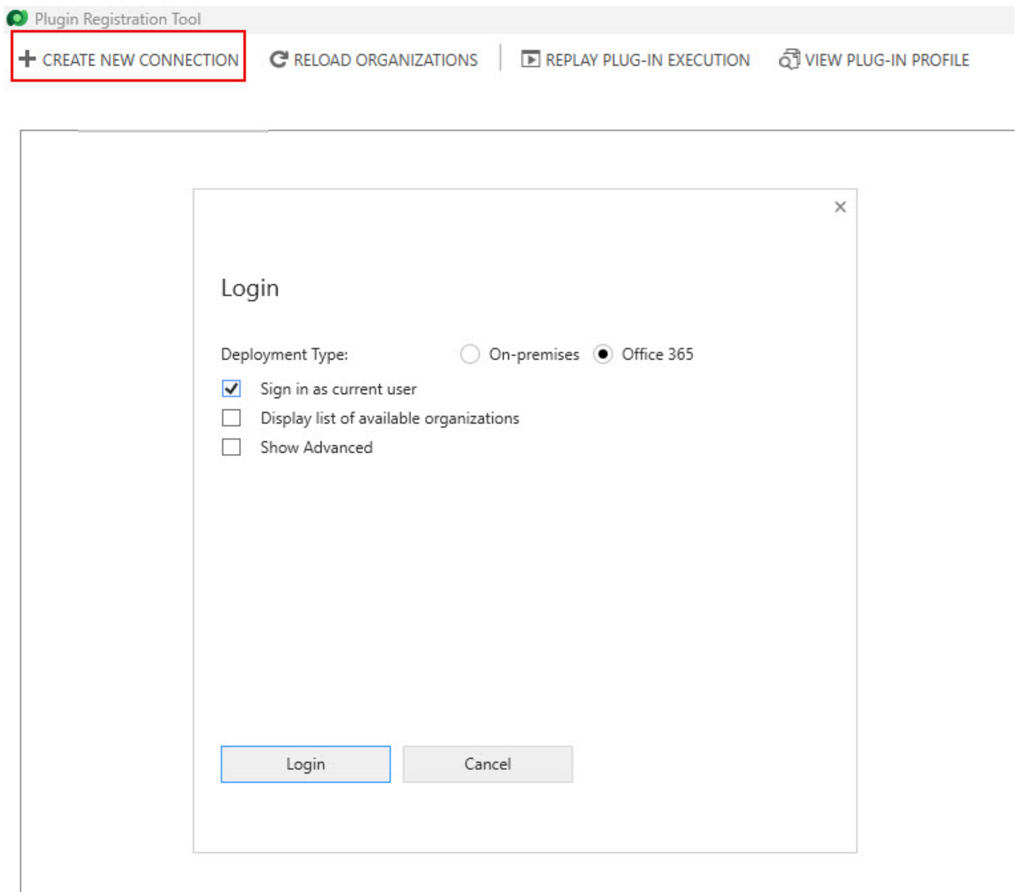
Webhooks are a way to communicate when an event occurs in one system with another system.

You can create webhooks in various ways. In this recipe, we have used the Plug-in Registration Tool.

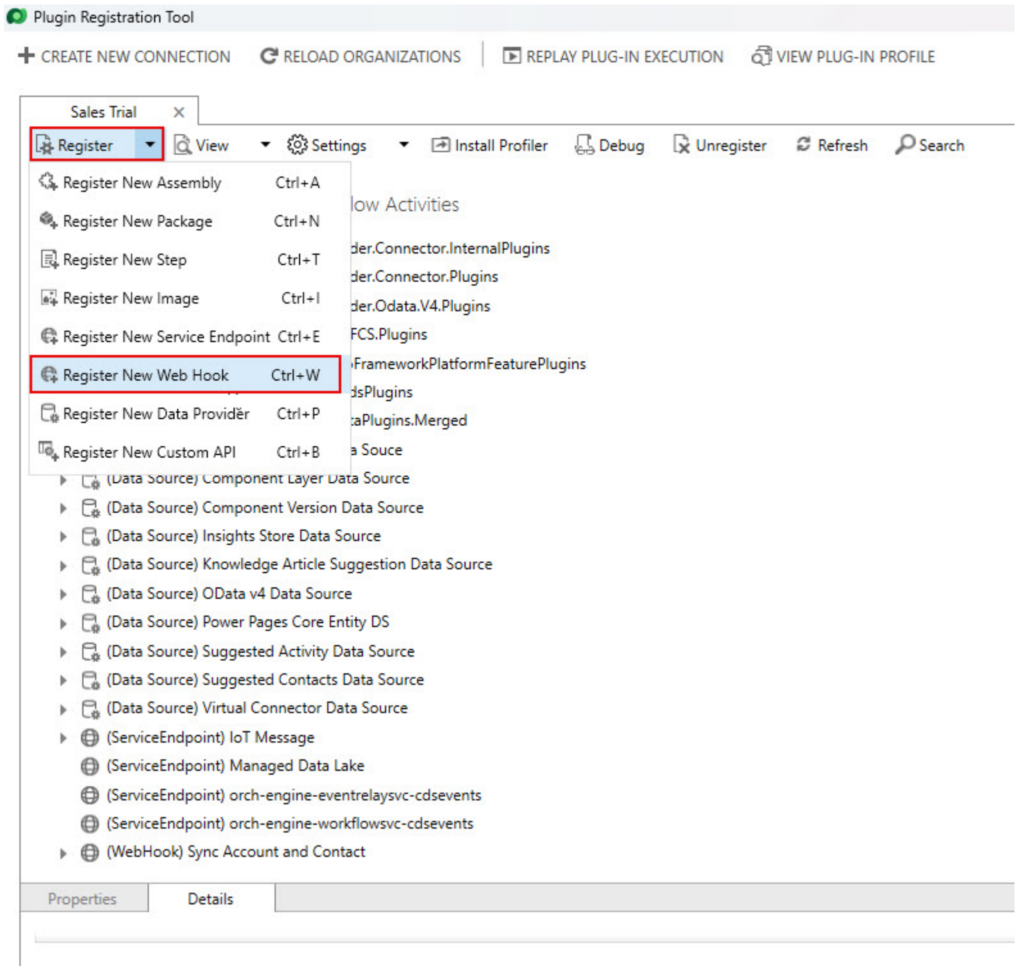
Use the Plug-in Registration Tool to [register a webhook](). To get the Plug-in Registration Tool, see [Dataverse development tools]().

To create a webhook in Dynamics 365, perform the following steps:

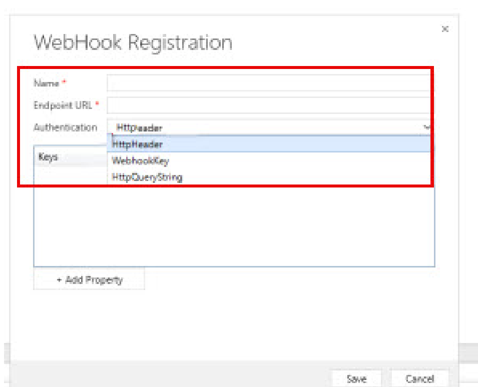
1. Open the Plug-in Registration Tool and log in to your organization.
2. Click **CREATE NEW CONNECTION** on the **Plugin Registration Tool** page.
   The following image shows the **CREATE NEW CONNECTION** button on the **Plugin Registration Tool** page:



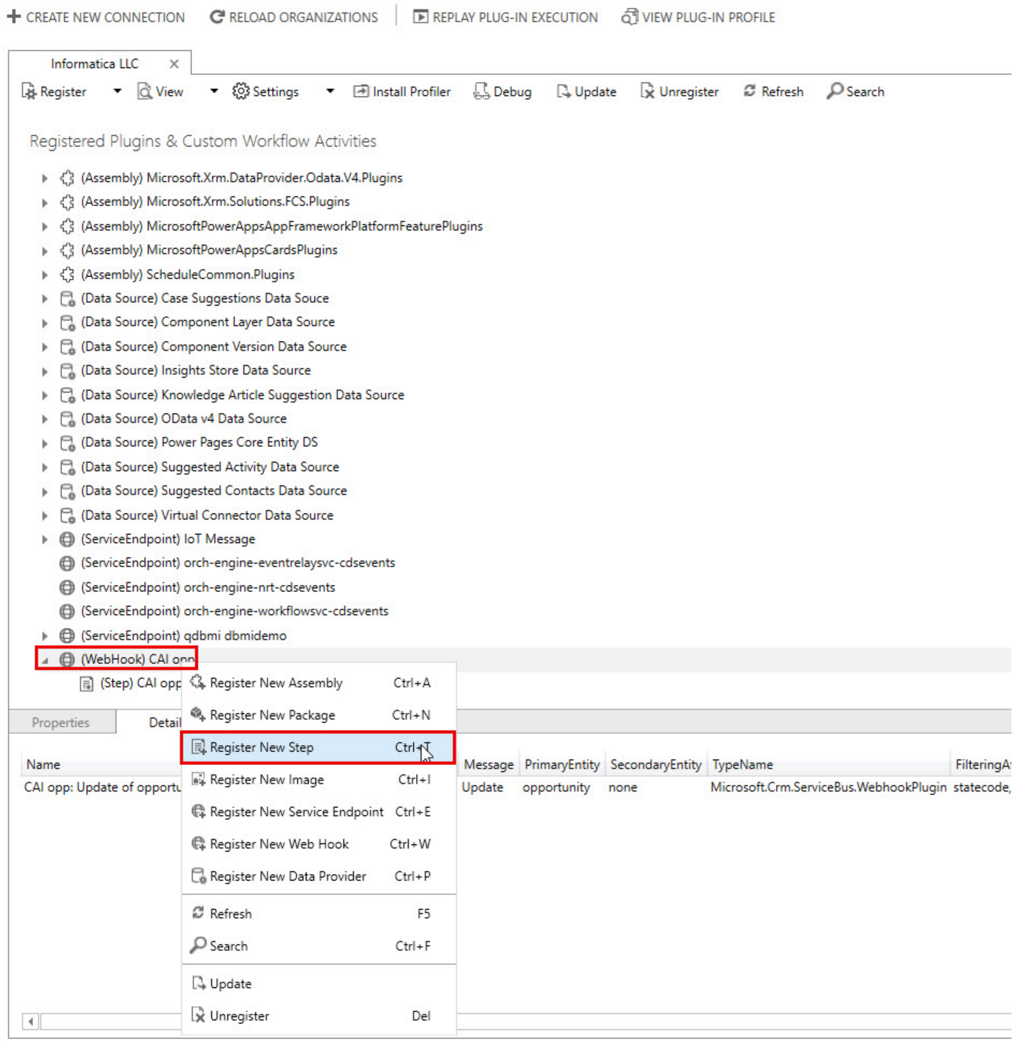
3. Go to **Register** and select **Register New Web Hook**.
   The following image shows the **Register New Web Hook** option on the **Register** menu:



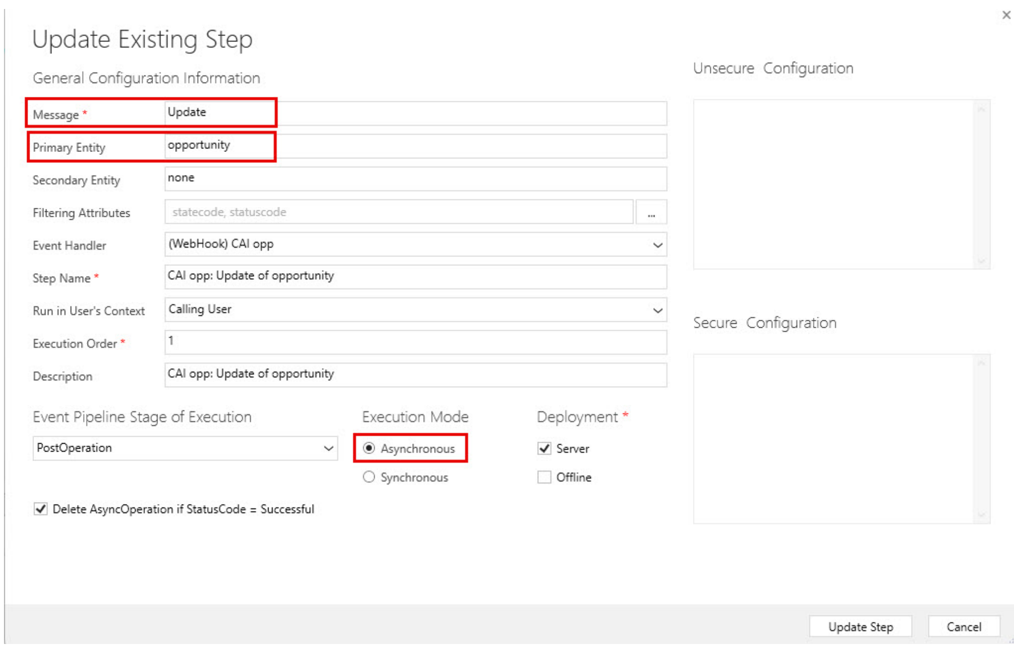
4. Enter the webhook name, the “Step 6: Invoke the process” on page 22 that you copied while invoking the process in the **Endpoint URL** field, and the authentication method in the **Authentication** field.
The following image shows the **WebHook Registration** dialog box:



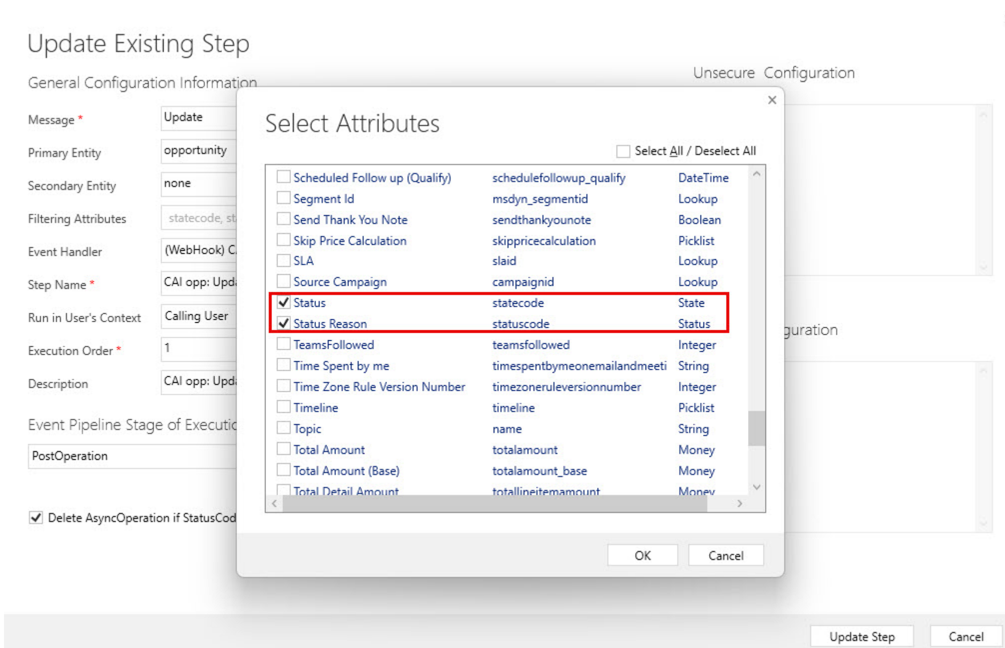
5. Go to **WebHook** and select **Register New Step**.
The following image shows the **Register New Step** option:



6. In the **Update Existing Step** page, enter the value as **Update** in the **Message** field, **opportunity** in the **Primary Entity** field, and select **Asynchronous** in the **Execution Mode** field.
   The following image shows the **Update Existing Step** page:



7. In the **Select Attributes** dialog box, clear all the fields and select the **Status** and **Status Reason** custom fields to trigger the webhook when the opportunity status is updated.
   The following image shows the **Status** and **Status Reason** custom field attributes in the **Select Attributes** dialog box:

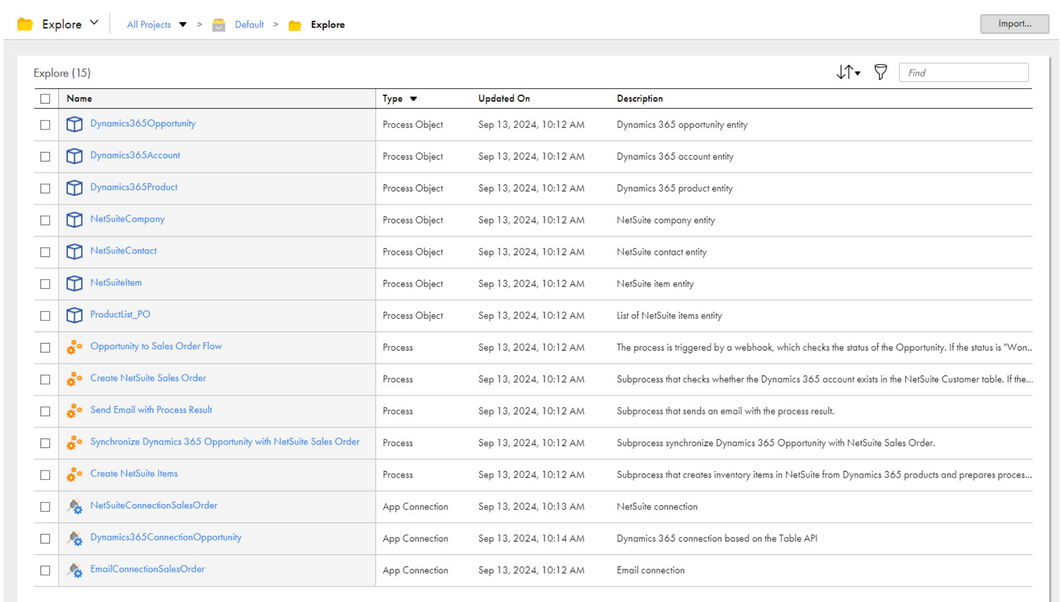


CHAPTER 2

Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe contents

The recipe contains multiple assets, such as process objects, app connections, and processes.

The following image shows the assets that the Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe package contains:



Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe assets

The following table lists the assets that the Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe package contains:

Asset Name	Asset Type	Description
Dynamics365Product	Process object	Dynamics 365 product entity.
Dynamics365Account	Process object	Dynamics 365 account entity.
NetSuiteItem	Process object	NetSuite item entity.
ProductList_PO	Process object	List of NetSuite items entity.
NetSuiteCompany	Process object	NetSuite company entity.
NetSuiteContact	Process object	NetSuite contact entity.
Dynamics365Opportunity	Process object	Dynamics 365 opportunity entity.
Dynamics365ConnectionOpportunity	App connection	Connects to Dynamics 365 based on the Table API.
NetSuiteConnectionSalesOrder	App connection	NetSuite connection.
EmailConnectionSalesOrder	App connection	Email connection.
Create NetSuite Sales Order	Process	Subprocess that checks whether the Dynamics 365 account exists in the NetSuite Customer table. If the account does not exist, the process creates a customer based on the account data. The process then checks if a sales order has already been created based on the Dynamics 365 opportunity. If a sales order exists, the user is notified. Otherwise, the process creates a new sales order and provides a confirmation message.
Create NetSuite Items	Process	Subprocess that creates inventory items in NetSuite from Dynamics 365 products and prepares process objects for creating sales orders.
Synchronize Dynamics 365 Opportunity with NetSuite Sales Order	Process	Subprocess that synchronizes Dynamics 365 opportunity with NetSuite sales order.

Asset Name	Asset Type	Description
Send Email with Process Result	Process	Subprocess that sends an email with the process result.
Opportunity to Sales Order Flow	Process	The process is triggered by a webhook to check the status of the Dynamics 365 opportunity. If the status is Won, the process creates sales orders based on the opportunity. When the sales order is created, the process sends an email with the results.

CHAPTER 3

Using the Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe

To use the Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe, you must perform the following steps manually:

Step 1: Copy and access the recipe

Step 2: Configure and publish the Email connection

Step 3: Configure and publish the Dynamics 365 connection

Step 4: Configure and publish the NetSuite connection

Step 5: Configure and publish the processes

Step 6: Invoke the process

Step 1: Copy and access the recipe

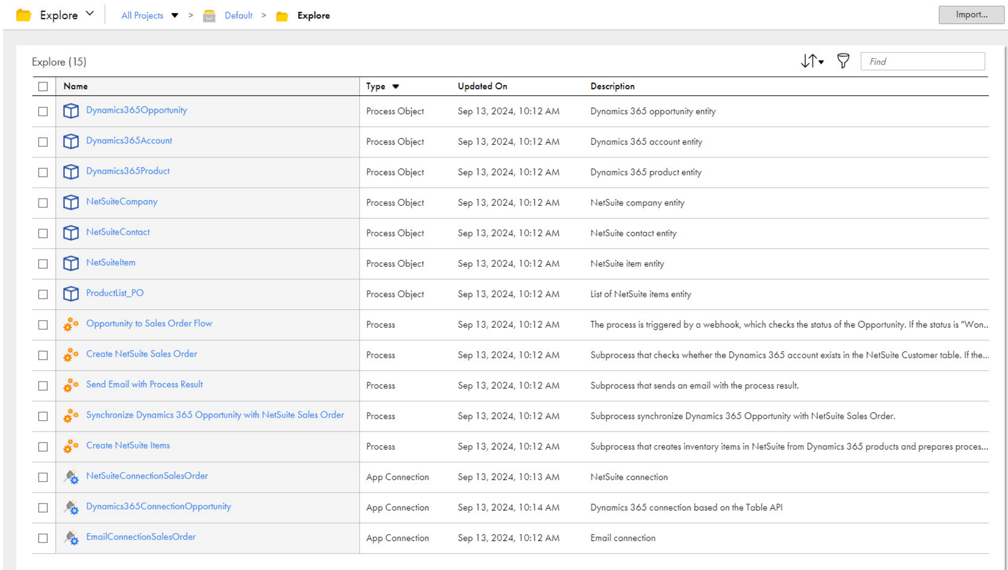
Copy the pre-configured assets in the recipe to a separate project or folder.

1. Open the **Create NetSuite Sales Orders from Dynamics 365 Opportunities** recipe and click **Use**.
2. Select the location where you want to copy the recipe, and then click **Continue**.
3. In the **Copying the recipe** dialog box, click **OK**.

   It might take some time for the recipe to get copied. You will receive a notification when the recipe is ready for use.
4. After the recipe is copied, click **Explore** to access the recipe content.

5. Navigate to the project or folder where you copied the recipe or enter the recipe name in the **Find** box.

All the assets in the recipe are displayed as shown in the following image:



Step 2: Configure and publish the Email connection

Configure the authentication type and authentication details in the EmailConnectionSalesOrder connection, and then publish the connection.

1. Open the **EmailConnectionSalesOrder** connection.
2. From the **Type** list, select **IICS Cloud Application Integration Email Service (Licensed for use)**.
3. From the **Run On** list, select **Cloud Server or any Secure Agent**.
4. From the **Authentication Type** list, select **Password** or **OAuth** as needed. Based on the authentication type selected, perform one of the following steps:
   - For **Password** authentication, enter values for the following properties in the **Connection Properties** section:

Property	Description
Authentication	Select **Enable**. Email Connector authenticates the user name and password that you enter in the email connection properties.
User Name	User name to log in to the email server. The user name is either the account name or the email address that is used to send the email with the synchronization results. For example: `notifyme@mydomain.com`

Property	Description
Password	Password for the email address.
Security	Select **SSL** for the Email connection to use the SSL protocol.

Configure the following common properties on the connection creation page:

Property	Description
Host	Email server's DNS name, such as `mail.mydomain.com`, or an IP address, such as `192.168.1.1`.
Port	Port for communication between the Process Server and the email server. Default is **25**.

- For **OAuth** authentication, enter values for the following properties in the **Connection Properties** section:

Property	Description
Authorization URL	Enter the OAuth authorization URL for the email service that is used to authorize the user request. For example: `https://login.microsoftonline.com/xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/oauth2/v2.0/authorize`
Token Request URL	Enter the OAuth token request URL that handles token requests. For example: `https://login.microsoftonline.com/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/oauth2/v2.0/token` The refresh token expires in 90 days. The user must authenticate again and publish the connection before the token expires.
Client ID	Specify the identifier value from the OAuth provider.
Client Secret	Enter the client secret to connect to the email application.
Scope	Specify the scope. The scope in OAuth authentication limits an application's access to a user's account. You can select multiple scopes for a single client. To enter multiple scopes, separate each value with a space. For a Microsoft Outlook email account, enter the following scope: `https://outlook.office.com/SMTP.Send offline_access`

Configure the following common properties on the connection creation page:

Property	Description
Host	Email server's DNS name, such as `mail.mydomain.com`, or an IP address, such as `192.168.1.1`.
Port	Port for communication between the Process Server and the email server. Default is **25**.

5. Save and publish the connection.

Step 3: Configure and publish the Dynamics 365 connection

Configure the tenant and client details in the Dynamics365ConnectionOpportunity connection, and then publish the connection.

1. Open the **Dynamics365ConnectionOpportunity** connection.
2. From the **Type** list, select **Dynamics365**.
3. From the **Run On** list, select **Cloud Server or any Secure Agent**.
4. In the **Connection Properties** section, enter values for the following properties:

Property	Description
Tenant_ID	Dynamics 365 tenant ID to get the access token. Enter the tenant ID that you generated under **Microsoft Entra ID** > **App registrations** in Dynamics 365 after creating the client credentials.
Client_ID	Dynamics 365 client ID to generate a valid access token. Enter the client ID that you generated under **Microsoft Entra ID** > **App registrations** in Dynamics 365.
Client_Secret	Dynamics 365 client secret that you generated under **Microsoft Entra ID** > **App registrations** in Dynamics 365.
Grant_type	Grant type that the Dynamics 365 instance uses to get an access token for third-party client authorization. Enter the value as **client_credentials**.
Resource_URL	URL to access the Dynamics 365 instance.

5. Save and publish the connection.

Step 4: Configure and publish the NetSuite connection

Configure the client ID, instance URL, certificate ID, and private key in the NetSuiteConnectionSalesOrder connection, and then publish the connection.

1. Open the **NetSuiteConnectionSalesOrder** connection.
2. From the **Type** list, select **NetSuite**.
3. From the **Run On** list, select **Cloud Server or any Secure Agent**.

4. In the **Connection Properties** section, enter values for the following properties:

Property	Description
Client ID	NetSuite OAuth 2.0 client ID to generate a valid access token. Enter the client ID that you generated from the **Integration** page in NetSuite.
Certificate ID	NetSuite certificate ID that you generated using **OAuth 2.0 Client Credentials** under **Setup** > **Integration** > **Manage Authentication** > **OAuth 2.0 Client Credentials** (M2M) Setup in NetSuite.
Instance URL	NetSuite applications suite instance URL. Enter the instance URL in the following format: `https://[accountid].suitetalk.api.netsuite.com`
Certificate Private Key	NetSuite certificate private key. Enter the PKCS8 certificate as a Base64-encoded string in the following format: `-----BEGIN PRIVATE KEY----- ..... -----END PRIVATE KEY-----`

5. Save and publish the connection.

Step 5: Configure and publish the processes

Configure the allowed user or group for authorization, email ID for sending the notification, and subsidiary ID to create a customer in NetSuite, and publish the process.

1. To publish the following processes, click **Actions** in the row that contains the processes in the order specified and select **Publish**.
    1. Create NetSuite Items
    2. Create NetSuite Sales Order
    3. Synchronize Dynamics 365 Opportunity with NetSuite Sales Order
    4. Send Email with Process Result
2. Open the **Opportunity to Sales Order Flow** process.
3. On the **Start** tab of the Start step, in the **Allowed Users** or **Allowed Groups** field, enter the user or group that has access to the process service URL at run time.
4. On the **Assignment** tab of the **Set Flow Configuration** step, enter values for the following fields:
    - In the **Email** field, enter the email ID to send the email notifications.
    - If your NetSuite environment uses a Subsidiary, in the **NetSuite_Subsidiary_ID** field, enter the subsidiary ID to create a customer in NetSuite.
5. Save and publish the process.

Step 6: Invoke the process

When you invoke the Opportunity to Sales Order Flow process, the process creates a NetSuite sales order from a Dynamics 365 opportunity based on the webhook request from Dynamics 365.

1. Open the **Opportunity to Sales Order Flow** process and click **Actions** > **Properties Details** > **Copy Service URL**.
2. Open a text editor and add the input fields and values to the service URL as shown in the following format:
   ```
   <Cloud Application Integration POD URL>/active-bpel/public/rt/<API_name>?
   webhookRequest=<Webhook request from Dynamics 365>
   ```
3. Open a browser and paste the service URL.

You can also invoke the process using the Run Using option, REST or SOAP API endpoints in API clients, such as cURL, Postman, SOAP UI, or through any programming language.

You must copy the service URL to the webhook configuration in "Creating a webhook in Dynamics 365" on page 9 tool. The process will be triggered automatically when an opportunity is closed in Dynamics 365.

Rules and guidelines

Consider the following rules and guidelines when working with the Create NetSuite Sales Orders from Dynamics 365 Opportunities recipe:

You must update the process according to your Dynamics 365 and Netsuite service configurations as there are many possible limitations based on different services business rules. You can find the some limitations in this topic that have been defined for the objects synchronized in the recipe process.

- The **Subsidiary** field for both the company and items must be identical. This enforces a restriction on the currency used. The currency must be the same as the currency specified in the Subsidiary settings.
- Discounts and taxes are not synchronized from Dynamics 365 because these fields are configured separately for the company and are applied based on the settings within NetSuite.
- When creating a sales order, the process passes the exchange rate. However, you must check if the NetSuite settings allow you to modify this value.