Oracle® Cloud

Using Recipes and Accelerators in Oracle Integration 3

G28326-02
May 2025

Oracle Cloud Using Recipes and Accelerators in Oracle Integration 3,

G28326-02

Primary Author: Oracle Corporation

Contents

Preface

Using Recipes and Accelerators in Oracle Integration 3 describes how to use recipes and accelerators in Oracle Integration.

Topics:

Audience

Using Recipes and Accelerators in Oracle Integration 3 is intended for users who want to use recipes and accelerators in Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- Oracle Integration documentation on the Oracle Help Center.

- Oracle Cloud at http://cloud.oracle.com.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Get Started

Oracle Integration provides a rich collection of run-ready, preassembled integration solutions known as **recipes** and **accelerators**. Use recipes and accelerators to jump-start building integrations.

Topics:

Learn About Recipes and Accelerators

Recipes and accelerators, collectively known as *prebuilt integrations*, are pre-assembled integration solutions. They are a great way to kick start your journey of building integrations for specific business scenarios.

Recipes are sample use cases that can be used as templates to give you a head start in building integrations. **Accelerators** are run-ready business integrations or technical patterns of large scale that can solve complex business problems. If you want to implement a specific business process, we recommend that you start by exploring our extensive list of recipes and accelerators in Oracle Integration. More likely than not, you will find a use case that solves your business problem.

Recipes and accelerators have all the resources required for specific business scenarios. The resources include integration flows, connections, lookups, certificates, libraries, and more. Using recipes and accelerators, you can quickly get started without having to build these resources from ground up. You can start with a recipe or accelerator, and then customize it to fit your business needs and requirements. See Configure Resources.

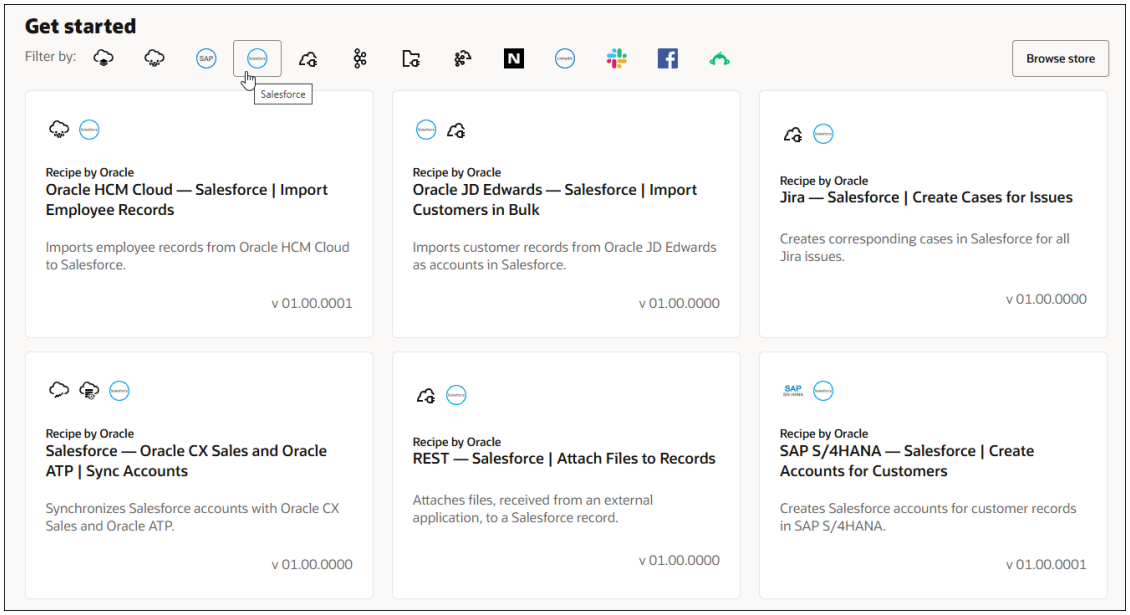
Here's a comparison of recipes and accelerators.

Recipes	Accelerators
Sample use cases that can be used as templates for building integrations. Many recipes are a good starting point to try out new features delivered in Oracle Integration. For example, explore the Import Files from File Server to OCI Object Storage recipe or the Manage Objects in OCI Object Storage recipe to understand about the OCI Object Storage native action.	End-to-end business processes or use cases (such as, marketing to lead, hire to retire, concept to launch) and technical solutions (such as sending alerts for failures) that solve specific business problems.
Can be edited in the Oracle Integration designer.	Can be configured and extended as per your business needs. See Extend an Integration in an Accelerator Project.
Cannot auto-upgrade to new version.	Can upgrade to a newer version (if available).

Find Recipes and Accelerators

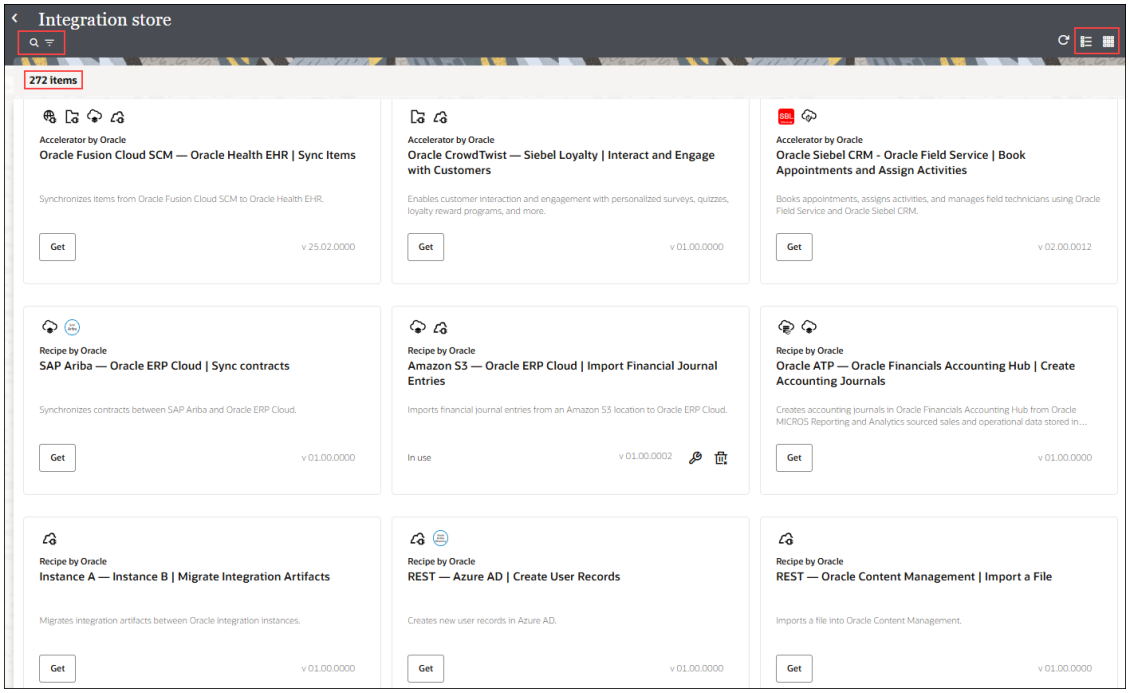
You can find a partial listing of the available accelerators and recipes under the **Get started** section of the Oracle Integration Home page. The Get started section also provides access to the **Integration store**, where you can browse the entire portfolio of preassembled solutions which includes recipes, accelerators and third-party adapters.

In the Get started section, you can filter the partial listing of recipes and acclerators according to applications used by clicking the product icon displayed on top. For example, to see some recipes and accelerators that use Salesforce, click the **Salesforce** icon.



To see all recipes and accelerators available in Oracle Integration, click **Browse store**. The Integration store opens displaying all artifacts hosted in it that includes recipes, accelerators, and third-party adapters. The total count of artifacts is displayed on top left of the Integration store. The count includes recipes and accelerators developed by Oracle, third-party recipes, third-party accelerators, and third-party adapters. Note that the third-party artifacts (recipes, accelerators and adapters) are also listed on Oracle Cloud Marketplace.

> **Note:**
>
> A third-party recipe, accelerator or adapter is built by an Oracle partner or an independent software vendor.



By default, all recipes, accelerators, and adapters in the Integration store are displayed in card view. To toggle between card and list view, use the **Card View** and **List View** icons on the Integration store.

Search recipes and accelerators

In the Integration store, click **Search** and enter the name of the recipe or accelerator in the field to search for a specific recipe or accelerator. Alternatively, you can also do a search based on keywords. To narrow down your search, click **Filter** next to the search field and apply filters based on **Type**, **Sort by** and **Built by**.

Get More Details About Recipes and Accelerators

The **Integration store** displays useful information to help you decide whether you want to use a particular recipe or accelerator.

At a glance, you can see:

- The adapters used by the recipe or accelerator, identified by product icons. Position the cursor over the icons to see the name of an adapter.
- Information about who built the recipe or accelerator.
- The two main applications that the recipe or accelerator connects to, followed by a brief purpose of the recipe.

Additionally, you may see different labels/buttons on the recipe or accelerator.

Label/Button	Description
Get	Indicates that the recipe or accelerator hasn't been installed. Click **Get** to install the recipe or accelerator.
In use	Indicates that the recipe or accelerator is installed. You can proceed to configure its resources, and then activate and run it.

Label/Button	Description
Get latest	Only available for accelerators. Indicates that a latest version of the accelerator is available. Click **Get latest** to install the latest version.
Go to marketplace listing	This label is displayed in recipes or accelerators developed by third parties. Clicking **Go to marketplace listing** will take you to the artifact listed on Oracle Cloud Marketplace.
Unlock	This label is displayed in third-party accelerators that can be installed only with access keys. Clicking **Unlock** opens the Unlock accelerator pane. To install the accelerator - enter the access key for the accelerator, accept the terms and conditions, and then click **Get**.

2

Install and Configure

Learn how to install a recipe or accelerator and then configure the connections and other resources within it.

Topics:

Install a Recipe or Accelerator

On your Oracle Integration instance, install the recipe or accelerator to deploy and configure the integrations and associated resources.

1. On the Oracle Integration Home page, in the **Get started** section, click **Browse store**.
2. Find the recipe or accelerator you want to install. See Find Recipes and Accelerators.
3. Click **Get** on the recipe or the accelerator card.

   A message confirms that the recipe or accelerator was successfully installed, and the recipe or accelerator card shows **In use**.

After you've installed the recipe or accelerator, configure its resources. See Configure Resources.

Uninstall a Recipe or Accelerator

Uninstalling a recipe or accelerator deletes it and all its resources.

You can't uninstall a recipe or accelerator if an integration flow of the recipe or accelerator is in the **ACTIVE** or **LOCKED** state.

Note that when you uninstall a recipe, you'll lose all the changes you made to the recipe's resources, including changes made to integration flows. Although you can reinstall the recipe, the resources are installed freshly in this case, without your modifications. If you'd like to make changes to an installed recipe, you can edit its resources instead of uninstalling it completely.

To uninstall a recipe or accelerator:

1. Find the recipe or accelerator that you want to uninstall. See Find Recipes and Accelerators.
2. Hover over the recipe or accelerator card and click **Uninstall**  .
3. Select which version you want to uninstall, then click **Uninstall**.

Configure Resources

After you've installed a recipe or accelerator, you need to configure its resources, including connections, integration flows and lookups.

Note that when you install a recipe or accelerator, it gets installed as a *project*. You can configure resources of a recipe or accelerator's project either from the Configuration Editor page using a guided wizard or directly from the project workspace. See About Projects in *Using Integrations in Oracle Integration 3*.

To configure resources of a recipe or accelerator using the wizard in the Configuration Editor, click **Configure** on the recipe or accelerator. See Edit Dependent Resources at the Project Level in *Using Integrations in Oracle Integration 3*.

To configure resources of a recipe or accelerator from the project workspace:

1. In the Oracle Integration navigation pane, click **Projects**.
2. In the Projects page, click the project name or click **Edit** .

   The project workspace opens displaying all the resources of the recipe or accelerator.
3. Within a section (for example, **Connections**), click **Actions** to view the options available for configuring the resource.

Option in the Action menu	Available for	Description
Add schedule	Integrations	Define when the integration flow runs. This option is available only if the integration flow uses a Scheduled Orchestration style. See Define the Integration Schedule in *Using Integrations in Oracle Integration 3*.
Delete	Connections Certificates Lookups Libraries	Delete an existing resource. You can't delete and replace a resource if an integration flow in the recipe is either active or locked. You can't delete and replace resources if they are included in an accelerator.
Edit	Connections Certificates Integrations Lookups Libraries	Access the corresponding page of the resource for editing a resource in a recipe or accelerator. For more information about editing these resources, see the following in *Using Integrations in Oracle Integration 3*: • Configure Connection Properties and Security Properties • Manage Security Certificates • Modify an Integration • Map Data and Manage Lookups • Use Libraries to Manage Integration Functions

Option in the Action menu	Available for	Description
Extend **Note:** Available only in accelerators.	Integrations	Extend an integration flow in an accelerator to process predefined custom objects by adding and configuring an extension group. See Extend an Integration in an Accelerator Project in *Using Integrations in Oracle Integration 3*.
Clone	Connections Integrations Lookups	Clone an existing connection, integration or lookup. For more information, see the following in *Using Integrations in Oracle Integration 3*: • Clone a Connection • Clone an Integration • Clone a Lookup
Create new version	Integrations	Create a new version of an integration. See Create a Draft of an Integration in *Using Integrations in Oracle Integration 3*.
Refresh endpoints	Integrations	Refresh the endpoints of an integration flow in a project that includes endpoints that support refreshing artifacts. See Refresh Endpoints of an Integration in a Project in *Using Integrations in Oracle Integration 3*.
Refresh metadata	Connections	Refresh the currently-cached metadata available to adapters that have implemented metadata caching. See Refresh Integration Metadata in *Using Integrations in Oracle Integration 3*.
Update property values	Integrations	Update integration property values for any integration flow in which properties have been defined.
View	Connections Integrations Lookups	View details about connections, integrations and lookups used in a recipe or accelerator.

4. Make the necessary edits to the resources and save your changes, then click **Go back** to return to the project workspace.

Add a resource

You can also add a new resource in your recipe or accelerator project. In the respective resource section, click **Add** if no resources currently exist or **+** if resources already exist.

For more details on how to add various resources such as connections, integrations and lookups in your project, check out the relevant information in *Using Integrations in Oracle Integration 3*.

- Create an Integration
- Create a Connection.

  Note that connection property and security values are specific to each adapter that is used in the recipe or accelerator.
- Create a Lookup

After you've configured the resources, activate and run the recipe or accelerator.

3
Activate and Monitor

After you've configured the resources in a recipe or accelerator, you can activate, run and monitor the integration flows of the recipe or accelerator.

- Activate a Recipe or Accelerator
- Monitor Integrations

Activate a Recipe or Accelerator

After configuring the resources of a recipe or accelerator, activate it and set the tracing levels for its integrations.

To activate a recipe or accelerator:

1. In the project workspace, click **Activate**.
2. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.

   To know more about tracing options while activating, see Activate an Integration in *Using Integrations in Oracle Integration 3*.

   A message confirms that the integrations have been activated. Refresh the page to view the updated status of the integrations.

The integrations of the recipe or accelerator are now available for you to run and test. To know more about how to run and test integrations, see Workflow for Testing Integrations in *Using Integrations in Oracle Integration 3*.

Monitor Integrations

After running the integrations within a recipe or accelerator, you can monitor the integration flows and manage errors (if any) in your project.

1. In the project workspace, click **Observe**. You'll see the integration flow/integration flows being triggered and running successfully.
2. To manage errors in your project, see Manage Errors in a Project in *Using Integrations in Oracle Integration 3*.

To know more about monitoring integrations, see Workflow for Monitoring Integrations in *Using Integrations in Oracle Integration 3*.

4

Explore Recipes and Accelerators

In this guide, you'll find an extensive list of recipes and accelerators that you can use in Oracle Integration.

However, the number of recipes and accelerators in Oracle Integration is constantly growing. For a full list of recipes and accelerators developed by Oracle, see Recipes and Accelerators on the Oracle Help Center. To get an even more comprehensive list of recipes and accelerators developed by Oracle as well as by third parties, browse the **Integration store** in Oracle Integration.

> **Note:**
>
> Oracle provides these recipes and accelerators as samples only. These recipes and accelerators are meant only for guidance, and are not warranted to be error-free. No support is provided for these recipes and accelerators.

Browse All

Browse all recipes and accelerators in this guide:

- Attach Files to Salesforce Records
- Calculate Total Balance
- Copy Objects from OCI Object Storage to Microsoft SharePoint
- Create Accounting Journals in Oracle Accounting Hub
- Create Accounts in Oracle CX Sales and B2B Service for HubSpot Companies
- Create Activities in Oracle Field Service Cloud for Oracle ERP Cloud Work Orders
- Encrypt and Decrypt Data Using OCI Vault
- Export Employee Images from Oracle HCM Cloud to an FTP Server
- Export Employee Records from Workday to Azure Active Directory
- Handle Throttling Using Parking Lot Pattern
- Import a File into Oracle Content Management
- Import a Payment Request into Oracle ERP Cloud
- Import Files from File Server to OCI Object Storage
- Import Financial Journal Entries from Amazon S3 to Oracle ERP Cloud
- Import Financial Journal Entries from an FTP server to Oracle ERP Cloud
- Manage Objects in OCI Object Storage
- Manage ServiceNow Incidents
- Migrate Integration Artifacts Between Instances
- Post Slack Notifications for Completed SurveyMonkey Surveys

- Post Slack Notifications for New Marketo Leads
- Post Slack Notifications for Oracle CPQ Quote Approvals
- Send Automatic Replies to Emails with a Specific Subject Using Oracle AI
- Send Files from OCI Object Storage to Oracle ATP
- Send Compliance Documents from DocuSign to New ServiceNow Users
- Sync Leads Between Marketo and Salesforce
- Sync Opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite Orders
- Sync Oracle NetSuite Customers with Oracle CX Sales and B2B Service Accounts
- Sync Workday Employees with ServiceNow Users
- Sync Oracle NetSuite Items with QuickBooks Products
- Sync Oracle Service Cloud Incidents with Oracle CPQ Transactions

Attach Files to Salesforce Records

Use this recipe to attach files to a Salesforce record from an external application.

Topics:

About This Recipe

When this recipe receives a REST request containing a file from an external application or from another integration flow in Oracle Integration, it attaches the file to the specified record in Salesforce. The recipe contains an app-driven orchestration integration, which uses the standard REST Adapter and the Salesforce Adapter.

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe. Subsequently, you can activate the integration flow of the recipe and send a POST request (containing a file and the ID of the Salesforce record to which you want to attach the file) to the integration's endpoint URL from an external application. When triggered, the integration flow attaches the file received to the specified record in Salesforce. If your REST request doesn't contain a file, the integration flow isn't triggered.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

To access Salesforce using Oracle Integration and attach files to records, you must perform certain configuration tasks on your Salesforce instance.

You must create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

In addition, you must identify your Salesforce instance type and your current Salesforce API version. Finally, you must obtain the ID of the Salesforce record to which you want to attach files.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

> **Note:**
>
> The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See Toggle or switch between Lightning Experience and Salesforce Classic.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
   a. On the Salesforce Setup page, expand **Manage Users** under the Administer section in the left navigation pane.
   b. Click **Profiles**.
   c. On the Profiles page, click **New Profile**.
   d. On the resulting page:
      i. Select **Standard User** in the **Existing Profile** field.
      ii. Enter a name for the new profile, for example, `API Enabled`, and click **Save**.

      The new profile is now saved, and the Profile Detail page of the new profile is displayed.
   e. Click **Edit** on the Profile Detail page.
   f. On the Profile Edit page:
      i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
      ii. Scroll to the Standard Object Permissions section and perform the following actions.
         - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
         - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
         - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
         - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.

iii. Scroll to the end of the page and click **Save**.

2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

> **Note:**
>
> If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.

b. Click **New User** in the resulting page.

c. On the New User page:

i. Enter a first name and last name for the user, for example, `Integration User05`.

ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note down this user name.

iii. In the **User License** field, select **Salesforce**.

iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.

v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.

d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce.

You're redirected to the Salesforce instance to set a password for the new user account.

e. Set a password and note down the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

> **Note:**
>
> If you're shown the Lighting Experience UI, switch to the Salesforce Classic UI. See Toggle or switch between Lightning Experience and Salesforce Classic.

f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.

i. Stay signed in as the new user, and click the user name at the top of the page to open a menu.

ii. Click **My Settings** in the menu.

iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.

iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.

v. Click the **Reset Security Token** button.

A new security token is sent to the email address associated with the account. Note down the security token.

vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.

3. Identify your current Salesforce API version.
   a. On the Salesforce Setup page, scroll to the Build section in the left navigation pane.
   b. Click **Develop**, and then **API**.
   c. On the API WSDL page, click the **Generate Enterprise WSDL** link.

      The WSDL is displayed in a new browser tab, and your current API version is present in the second line. For example:

      ```
      Salesforce.com Enterprise Web Services API Version 52.0
      ```

   d. Note down the API version.
4. Identify your Salesforce instance type.
   a. If you use the URL https://login.salesforce.com to log in to your Salesforce account, your instance type is **Production**.
   b. If you use the URL https://test.salesforce.com to log in to your Salesforce account, your instance type is **Sandbox**.
5. Identify the Salesforce record to which you want to attach files and obtain the record's ID. As an example, we'll use an account record.
   a. On the Salesforce Setup page, click the **Accounts** tab.

      > **Note:**
      >
      > If you don't see the **Accounts** tab on the Setup page, click the **Plus** icon to the right of your current tabs, and then click **Accounts**.

   b. On the Accounts Home page, click an account to open it.
   c. In the account page's URL, note down the string after the last forward slash (/).

      This is the account record's Salesforce ID. You'll specify this ID while sending a REST request with the file attachment from an external application to the integration flow of this recipe.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Save**. If prompted, click **Save** again.

3. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

4. To return to the project workspace, click **Go back**.

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select **Production** or **Sandbox** based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Before You Install the Recipe.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Salesforce Username Password Policy** selected.
Username	Enter the user name of the account created for Oracle Integration on Salesforce. See Before You Install the Recipe.
Password	Enter the password of the account created for Oracle Integration on Salesforce. **Note:** To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator
2. Run the recipe from an external application.
   a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

b. On the Configure and run page, click **Endpoint metadata**.

c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.

d. From the external application, send a `POST` request to this endpoint URL along with the necessary file and the ID of the Salesforce record to which you want to attach the file. Upload the file in the `POST` request's **Body** field and specify the Salesforce record's ID as a template parameter at the end of the integration's endpoint URL, replacing the placeholder `{salesforceObjectId}`.

   The integration flow attaches the file you uploaded to the specified Salesforce record and returns **200 OK** as a response, which indicates a successful execution.

3. Test the recipe in Oracle Integration.

   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

   b. On the Configure and run page, in the Request section, enter the following details:

      i. On the **URI Parameters** tab, against the **salesforceObjectId** field, enter the ID of the Salesforce record to which you want to attach the file. To obtain the Salesforce ID of a record, see Before You Install the Recipe.

      ii. On the **Body** tab, upload a file from your computer.

   c. Click **Run**.

      The integration flow now attaches the file you uploaded to the specified Salesforce record. The Activity Stream pane appears displaying the status of the integration instance's execution.

   d. In the Response section of the test page, you should see the status as **200 OK** and **"success" : true**.

4. Log in to your Salesforce instance as an Administrator and check if the file has been attached to the specified record. Because we've used an account record as an example in this document, the following steps detail how to navigate to an account record on Salesforce and check for the file attachment.

   a. On the Salesforce Setup page, click the **Accounts** tab.

   > **Note:**
   >
   > If you don't see the **Accounts** tab on the Setup page, click the **Plus** icon to the right of your current tabs, and then click **Accounts**.

   b. On the Accounts Home page, click the account whose ID you specified in the REST request.

   c. On the account's page, scroll to the Notes & Attachments section.

      You'll find the file you sent through the REST request under this section.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Calculate Total Balance

Use this recipe to calculate the total balance of a group of credit cards.

Topics:

About This Recipe

This recipe demonstrates how you can use the for-each action to iterate through an array of credit cards, and calculate the total balance of the credit cards.

To use the recipe, you must install the recipe and configure the connection and other resources within it. The recipe contains the integration **Oracle SOAP Library Calculator ForLoop**. To run the integration, the input you provide must use WS-Security UsernameToken authentication and specify a user, a password, a token, and a time stamp. When the integration flow of the recipe is triggered, it receives an array of credit cards in the SOAP request. Subsequently, the recipe calculates and logs the total balance of all the credit cards in the array. It uses the JavaScript library Oracle-Library-Calculator to add up the total balance, and the logger action to log the result to the activity stream.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Oracle-Library-Calculator JS Library

Before You Install the Recipe

To create a connection to the SOAP service, perform the configuration steps described in Create a SOAP Adapter Connection.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources.

Configure the Oracle SOAP Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connection, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator
2. Run the recipe from an external application.
   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run Details**.
   b. In the panel that opens, copy the **Metadata URL**. This is the integration's endpoint URL.
   c. From the external application such as SoapUI, load the WSDL hosted at the **Metadata URL** and send a request to the available operation along with the request message data.
   d. Provide the request message data with card number, balance, loan amount and credit rate.

      For example:

      ```
      CARD NUMBER: 5105 1051 0510 5100

      BALANCE: 151

      LOAN AMOUNT: 100

      CREDIT RATE: 4
      ```

   e. Authenticate your request.

      The integration flow of the recipe is triggered.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations
4. Verify the result which should be the total balance of the credit cards.

Related Documentation

- *Using the SOAP Adapter with Oracle Integration 3*

Copy Objects from OCI Object Storage to Microsoft SharePoint

Use this recipe to copy objects from OCI Object Storage to Microsoft SharePoint.

Topics:

About This Recipe

This recipe demonstrates how you can move objects from OCI Object Storage to SharePoint, and then delete the original objects in OCI Object Storage.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When the integration flow of the recipe gets triggered, it copies objects from the specified OCI Object Storage bucket and uploads them to a SharePoint folder. The integration flow of the recipe uses a native action to connect to OCI Object Storage and the REST adapter to connect to SharePoint. SharePoint REST APIs are used to upload objects to SharePoint.

System and Access Requirements

- Oracle Integration 3
- An account on Oracle Cloud Infrastructure with access to create OCI Object Storage bucket
- An account on Microsoft SharePoint with access to upload files to a SharePoint folder

Before You Install the Recipe

To successfully connect to OCI Object Storage and SharePoint using Oracle Integration and copy OCI Object Storage objects into SharePoint, you must perform certain configuration tasks on your OCI Object Storage and SharePoint instances.

Configure OCI Object Storage

To successfully connect to OCI Object Storage using a native action, create a dynamic group and required policy.

See **Prerequisites** in Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action.

Additionally, create an OCI Object Storage bucket and upload objects into it. See Creating an Object Storage Bucket and Uploading an Object Storage Object to a Bucket.

Configure SharePoint

To successfully connect to SharePoint and upload OCI Object Storage objects, you must perform certain configurations on SharePoint.

1. Register the SharePoint add-in and get the client Id and secret.
   a. Go to: `https://<sharepoint domain>.sharePoint.com/sites/<sharepoint Site Name>/_layouts/15/appregnew.aspx` by using a web browser.
   b. In the AppRegNew page, enter the following details:

Field	Information to Enter
Title	Enter a title of your choice. For example: `OICAPP`
App Domain	Enter the local hostname.
Redirect URL	Enter the endpoint URL. For example, `https://localhost.com`

   c. Click **Create**.

   You get the client Id and client secret in the resulting window. Note down the client Id and secret. You'll need these while creating a connection to SharePoint from Oracle Integration.

   d. Provide permission to the add-in that you created.

i. Go to: `https://<sharepoint_domain>.sharepoint.com/sites/<sharepointSiteName>/_layouts/15/appinv.aspx`

ii. Enter the client Id in the **App Id** field and click the **Lookup** button. The **Title**, **App Domain**, and **Redirect URL** fields will be populated automatically.

iii. Enter the following in the **App's permission Request XML** text area.

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
    <AppPermissionRequest Scope="http://sharepoint/content/
sitecollection/web" Right="FullControl" />
</AppPermissionRequests>
```

iv. Click **Create**.

v. In the resulting window, click **Trust It**.

2. Get the SharePoint tenant Id.

a. Go to: `https://<sharepoint_domain>.sharepoint.com/_layouts/15/appprincipals.aspx`.

b. In the **Site Settings>Site Collection App Permissions** page, copy the alphanumeric string after @ in the **App Identifier** value.

This is the SharePoint tenant Id.

3. Get the SharePoint resource Id.

a. Enter the following `POST` endpoint in Postman.

`https://<sharepoint_domain>.sharePoint.com/_vti_bin/client.svc/`

b. Add the **Bearer** type Authorization header without entering any bearer token and click **Send**.

c. In Postman, go to the **WWW-Authenticate** response header and copy the value of the client id.

This is the SharePoint resource Id.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources.

Configure the SharePoint REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the **Connection URL**. For example: `https://<sharepoint_domain>.sharepoint.comm`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **OAuth Custom Two Legged Flow**.

Field	Information to Enter
Access Token Request	`-X POST -H "Content-Type: application/x-www-form-urlencoded" -d 'grant_type=client_credentials&client_id=<client_id>@<tenant_id>&client_secret=<client_secret>&resource=<resource_id>/<sharepoint_domain>.sharepoint.com@<tenant_id>' https://accounts.accesscontrol.windows.net/<tenant_id>/tokens/OAuth/2` Replace `client_id`, `client_secret`, `tenant_id` and `resource_id` in the above with respective values for the same that you obtained while configuring SharePoint. See Configure SharePoint.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Lookup Table

Edit the lookup table **Utility_Lookup** and set up appropriate values for the lookup keys as per your requirement.

1. In the Lookups section, click the lookup name.
2. Enter appropriate values for the lookup keys.

Key	Value
SP_Site_Name	Specify the SharePoint site name.
SP_Folder_Name	Specify the SharePoint folder name under which objects are to be uploaded.
OCI_Object_Storage_Bucket_Name	Specify the OCI Object Storage bucket name from which objects are to be imported to SharePoint.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** ... on the **Object Synch from OCI Object Storage to Sharepoint** integration flow, then select **Run**.
   b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The recipe now copies objects from OCI Object Storage and uploads them to SharePoint. It then deletes the original objects in OCI Object Storage.

> **Note:**
>
> You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations
4. Log in to your SharePoint instance and verify that the objects imported from OCI Object Storage are uploaded to the specified SharePoint folder.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Create Accounting Journals in Oracle Accounting Hub

Use this recipe to create accounting journal entries in Oracle Accounting Hub for transaction data received from external systems.

Topics:

About This Recipe

This recipe imports and transforms transaction data received from external systems and creates accounting journal entries for the same in Oracle Accounting Hub.

To use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe contains the following scheduled integration flows.

- **Oracle FTP ERP ImportAccountTransactions ESS Trigg** - This integration flow imports the transaction data from an FTP server and writes the data into an Oracle ATP database table. It fetches the data stored in the Oracle ATP database table and uploads it to Universal Content Management (UCM) using the Oracle ERP adapter. It then places a request to submit a Enterprise Scheduler Service (ESS) job and inserts the ESS job request id to the Oracle ATP database.
- **Oracle ERP AccountTransactions ESS Job Monitor** - This integration flow gets each request id from the Oracle ATP database table and queries for the ESS job status in Oracle ERP Cloud. If the status is successful, it sends the data to Oracle Accounting Hub where a corresponding accounting journal entry is created.

Prerequisite: For the recipe to run successfully, the transaction data received from external systems have to be uploaded to the FTP server in a particular format. Download the XlaTransactionUploadTemplate99.zip for standard transaction header, line and metadata file templates that can be used to upload the required transaction data (from external systems) to the FTP server.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- Oracle Accounting Hub
- An account on Oracle Accounting Hub with the Administrator role
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- Oracle ATP
- An account on Oracle ATP with the Administrator role
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server, Oracle ATP, Oracle ERP Cloud and Oracle Accounting Hub instances to successfully connect to Oracle Integration and create accounting journal entries.

Access Your FTP Server

Obtain an FTP server and ensure that you can access it.

1. Login to the server using your username and password through an FTP client; for example, FileZilla.
2. Create an input folder to store the transaction data files. Note down the path of the folder.
3. Upload the transaction data files into the input folder.

You can also choose to store the input transaction data files to work with on File Server, an embedded sFTP server within Oracle Integration. To use File Server, you must enable it for your Oracle Integration instance. See Enable File Server in *Using File Server in Oracle Integration 3*.

Configure Oracle ATP

Log in to your Oracle ATP database instance as an **Administrator** and perform the following tasks.

1. Perform general configuration tasks. See Prerequisites for Creating a Connection.
2. Create a schema named `accountinghub`.
3. Create a database table under the `accountinghub` schema.

```
Create Table V_JOBS_STATUS(RequestID VARCHAR2 (50),ERPStatus VARCHAR2
(50),Processed VARCHAR2 (50))
```

For more information, refer to Oracle Autonomous Database documentation.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign required roles to the user account.
    a. Create a new job role under role category Common - Job Roles.
    b. Under role hierarchy, add the following duties:
        - `ORA_XLA_ACCOUNTING_HUB_INTEGRATION_DUTY`
        - `ORA_XLA_ACCOUNTING_HUB_INTEGRATION_DUTY_OBI`
        - `FIN_FUSIONACCOUNTINGHUB_IMPORT_RWD`
    c. Save the job role.
    d. Assign the job role to a user.
    e. Run the following jobs:
        - Send Pending LDAP Requests
        - Retrieve Latest LDAP Changes
        - Import User and Role Application Security Data

Configure Oracle Accounting Hub

To configure Oracle Accounting Hub, you have to complete three primary steps.

1. Create a new sub-ledger application by registering the transaction source system using a spreadsheet.
2. Configure accounting rules.
3. Upload transaction data to create accounting entries.

For more information, refer to the following:

- The **Set up Accounting Hub Cloud** section in this blog.
- Accounting Hub Best Practices.
- Implementing Accounting Hub

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle ATP ImportAccountTransaction Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See Configure Oracle ATP.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **JDBC With OCI Signature**.
Wallet	Select the check box, then click **Upload** to upload the wallet file. See Configure Oracle ATP.
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.
Object Storage Region	Specify the region in which your object storage is located. For example, `us-ashburn-1`.
Object Storage Tenancy OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, `ocid1.tenancy.oc1.`*`alphanumeric.value`*
Object Storage Compartment OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, `ocid1.compartment.oc1.`*`alphanumeric.value`*
Object Storage User OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, `ocid1.user.oc1.`*`alphanumeric.value`*
Private Key	Click **Upload** to select the key you created. Ensure that the key is in RSA (PKCS1) format. **Note:** Only a private key *without* a pass phrase/ password is supported.
Finger Print	Enter the finger print that was generated when you created the key in the Oracle Cloud Infrastructure Console.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle ERP SOAP ESS ImportAccountTrans Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the **WSDL URL**. For example: `https://<ERP Hostname>/publicFinancialCommonErpIntegration/ErpIntegrationService?wsdl`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the user name.
Password	Enter the password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle FTP ImportAccountTransactions Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter the port of your sFTP server.
SFTP Connection	Select **Yes** from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **FTP Server Access Policy**.
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle ERP ImportAccountTransactions Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud.
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** on the **Oracle FTP ERP ImportAccountTransactions ESS Trigg** integration flow, then select **Run**.

      The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

      - **jobDefinitionName**: Enter the name of the ERP ESS job. For example, `XLATXNIMPORT`.
      - **jobPackageName**: Enter the ERP ESS package name. For example, `/oracle/apps/ess/financials/subledgerAccounting/shared`

   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

   c. In the Integrations section of the project workspace, click **Actions** on the **Oracle ERP AccountTranactions ESS Job Monitor** integration flow, then select **Run**.
   d. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

> **Note:**
>
> You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. In Oracle Accounting Hub, check if a corresponding accounting journal entry is created for the transaction data that was uploaded to the FTP server.

Related Documentation

- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*
- *Using the SOAP Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Create Accounts in Oracle CX Sales and B2B Service for HubSpot Companies

Use this recipe to create sales accounts in Oracle CX Sales and B2B Service for companies in HubSpot.

Topics:

About This Recipe

This recipe creates accounts in Oracle CX Sales and B2B Service for companies in HubSpot as per a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the company records in HubSpot and, in turn, creates corresponding sales accounts in your Oracle CX Sales and B2B Service instance. Additionally, if a company record is updated, the recipe updates the corresponding sales account in Oracle CX Sales and B2B Service.

Basic data associated with companies in HubSpot, such as company domain name, organization name, postal address, phone, and so on are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- HubSpot
- An account on HubSpot with the Administrator role

- Oracle CX Sales and B2B Service
- An account on Oracle CX Sales and B2B Service with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your HubSpot and Oracle CX Sales and B2B Service instances in order to successfully connect to these external systems using Oracle Integration and create accounts for companies.

Configure HubSpot

To configure HubSpot, create a company. See Create companies.

Also, add a property in the company record to store the Oracle CX Sales and B2B Service sales account party number. See Create and edit properties. Specify the name of the property as `b2bpartynumber`.

Configure Oracle CX Sales and B2B Service

To access the Oracle CX Sales and B2B Service instance from Oracle Integration, you'll require a separate user account on Oracle CX Sales and B2B Service.

Perform the following tasks to configure Oracle CX Sales and B2B Service for this recipe.

1. Create a new user account for Oracle Integration. You'll use the credentials of this user account to connect to Oracle CX Sales and B2B Service from Oracle Integration.

   For the procedure to create the integration user account, see Creating an Integration User Account in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
2. Assign the following roles to the user account:
   - `Integration Specialist`
   - `AttachmentsUser`
   - `FND_MANAGE_CATALOG_SERVICE_PRIV`

   For steps to assign roles, see Assign Integration Roles in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
3. Make a note of the user name and password you set for the account.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle HubSpot Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Site URL	Enter `https://api.hubapi.com`.
REST Version	Enter your HubSpot version. For example, `v3`.

3. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the Client ID obtained when you configured HubSpot. See Configure HubSpot.
Client Secret	Enter the Client Secret obtained when you configured HubSpot.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

   You'll now be redirected to the HubSpot sign in page.
7. Enter your HubSpot account credentials.

   A page is displayed asking for permission to interact with the account.
8. Click **Yes**.

   You're informed that access is allowed. You can now switch back to the Oracle HubSpot Connection window of Oracle Integration to test your connection.
9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
10. To return to the project workspace, click **Go back**  .

Configure the Oracle B2B Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: `CX Sales and B2B Service Host`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the user name of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Configure Oracle CX Sales and B2B Service .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Configure the Lookup Table

Edit the **Country_Code_Lookup** lookup table to map the country codes between HubSpot and Oracle CX Sales and B2B Service.

1. In the Lookups section, click the lookup name.
2. In the **HubSpot** column, enter the name of the country you want to map.
3. In the **Oracle CX Sales and B2B Service** column, enter the country code to which the country in HubSpot has to be mapped.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Update property values for the integration flow. You can update the integration properties only after activating this flow.
   a. In the Integrations section, click **Actions** on the integration flow, then select **Update property values**.
   b. In the Update Property Values dialog, update the following:

Integration Property	Description
maxPageSize	Specify the maximum number of records per page
notificationMail	Specify the email address to which the error notifications are sent. Enter an email address of your choice

   c. Click **Submit**.

   A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** on the integration flow, then select **Run**.
   b. On the Configure and run page, click **Run**.

   You've now successfully submitted the integration for execution.

> **Note:**
>
> - You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.
> - If any record fails, an error notification email is sent to the user.

4. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
5. Log in to your Oracle CX Sales and B2B Service instance and check for the new sales accounts created.
   a. Log in to your Oracle CX Sales and B2B Service instance.
   b. From the menu, select **Sales** and then select **Accounts**. You will see the recent sales accounts created.

Related Documentation

- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
- *Using the HubSpot Adapter with Oracle Integration 3*

Create Activities in Oracle Field Service Cloud for Oracle ERP Cloud Work Orders

Use this recipe to create corresponding activities in Oracle Field Service Cloud when maintenance work orders are created in Oracle ERP Cloud.

Topics:

About This Recipe

This is an application-driven recipe that creates an activity in Oracle Field Service Cloud whenever a maintenance work order is created in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a new maintenance work order is created in the Oracle ERP Cloud instance, the recipe is triggered, and it creates a corresponding activity in Oracle Field Service Cloud. Basic data is synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- Oracle Field Service Cloud
- An account on Oracle Field Service Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Oracle Field Service Cloud instances in order to successfully connect to these external systems using Oracle Integration and create activities for maintenance work orders.

Configure Oracle Field Service Cloud

Retrieve the client ID and secret from Oracle Field Service Cloud by adding an application.

To add an application:

1. Log in to an Oracle Field Service Cloud instance.
2. In the top-left corner, click the hamburger menu, then click **Configuration**.
3. On the Configuration page, under **Subsystems**, click **Applications**.
4. Click **Add Application**.
5. Enter the **Application Name** and **Application ID**, give the required API access. See Authorize Access to APIs.
6. Click **Submit**.
7. Select the type of token service for the registered client application from the **Token service** drop-down list.

   By default, the type of token service is **Oracle Field Service**.
8. Select the authentication settings as **Authenticate using Client ID/Client Secret**.
9. Click **Show Client ID and Client secret** to copy them for configuring the connection in Oracle Integration.
10. Click **Save**.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Also, create the following two segments in the **Work Order Header** flexfield:

- `senttoofsc`: Indicates if the work order details are successfully sent to the Oracle Field Service Cloud instance.
- `OFSC SYNC STATUS`: Indicates if the new activity records are successfully created in the Oracle Field Service Cloud instance.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Create a Fusion Applications user to use in the Oracle ERP Cloud connection, and assign it the **Integration Specialist** role.

   This is the user you will specify in the ERP connection to connect to Oracle ERP Cloud.

   For more information, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
3. Update the **Work Order Header** flexfield with two new segments, `senttoofsc` and `OFSC SYNC STATUS`.

   a. In the top-right corner of the Oracle ERP Cloud home page, click the profile image, then select **Setup and Maintenance**.

b. On the right, click **Task**, and select **Search**.

c. In the Search field, enter `Manage Descriptive Flexfields`, and then click **Search**.

d. From the search results, select **Manage Descriptive Flexfields**.

e. On the Manage Descriptive Flexfields page, select **Work Orders** from the Module drop-down menu, then click **Search**.

f. From the search results, select **Work Order Header** and click **Edit**.

g. On the Edit Descriptive Flexfield page, click the **Create** icon in the Global Segments section. See Create Global Segments.

h. On the Create Segment page, enter the name for the segment as `senttoofsc`, and enter other values for the segment as required.

i. Repeat steps 3g and 3h to create another segment named `OFSC SYNC STATUS`.

j. Click **Save and Close**.

4. Deploy the descriptive flexfield. See Deploy Descriptive Flexfields.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle Field Service Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Field Service Cloud API URL	Enter the URL of your Oracle Field Service Cloud instance. See Configure Oracle Field Service Cloud.
Instance ID	Enter the ID of your Oracle Field Service Cloud instance. For example, `ofsc-xyz.test`.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Client Credentials**.
Client Id	Enter the client ID obtained from your Oracle Field Service Cloud instance. See Configure Oracle Field Service Cloud.
Client Secret	Enter the client secret obtained from your Oracle Field Service Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the username of the account created for Oracle Integration on the Oracle ERP Cloud instance. See Configure Oracle ERP Cloud.
Password	Enter the password of the account created for Oracle Integration on the Oracle ERP Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Lookup Table

Edit the **ORACLE-BRT-OFSC_CONFIG** lookup table to map **MntWorkOrderDefinitionCode** in Oracle ERP Cloud with the corresponding **OFSCActivityType** and **OFSCResource** in Oracle Field Service Cloud.

1. In the Lookups section, click the lookup name.
2. Map the maintenance work order definition codes in Oracle ERP Cloud with the corresponding activity type and resource in Oracle Field Service Cloud.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.

   Create a maintenance work order in Oracle ERP Cloud.

   a. Log in to your Oracle ERP Cloud instance.
   b. Under Supply Chain Execution, click **Maintenance Management**.
   c. Click the **Task** icon on the right and select **Manage Maintenance Work Orders**.

d. On the Maintenance Work Orders page, click **Create**.

e. In the Create Maintenance Work Order window, enter the details of the new work order and click **Save and Close**.

You've now successfully triggered the integration flow of the recipe.

When you create a maintenance work order in Oracle ERP Cloud, the **Oracle ERP FSC MWorkOrder To Activity Create Sync** integration flow gets triggered and a corresponding activity is created in the Oracle Field Service Cloud instance.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Check for the new activities created in your Oracle Field Service Cloud instance.
   a. Log in to the Oracle Field Service Cloud instance.
   b. Select **Dispatch Console** in the left navigation pane.
   c. Check if a corresponding activity is created for the maintenance work order created in Oracle ERP Cloud.

Related Documentation

- *Using the Oracle Field Service Cloud Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Encrypt and Decrypt Data Using OCI Vault

Use this recipe to encrypt and decrypt the field-level data by leveraging OCI Vault.

Topics:

About This Recipe

This recipe enables you to encrypt or decrypt data received from external systems using the OCI Vault. You can encrypt sensitive information (such as credit card details, salary information, and so on) before you pass it to target applications. Similarly, you can decrypt data from a third-party application and transfer it to another application.

This recipe uses the OCI Vault REST APIs to encrypt and decrypt the data through the REST adapter. It contains the following two integration flows:

- **Encrypt Data**: Accepts plain text as part of the request payload and invokes the OCI Vault Encrypt REST API, which encrypts the plain text and sends the cipher text as part of the REST API response.
- **Decrypt Data**: Accepts the cipher text as part of the request payload and invokes the OCI Vault Decrypt REST API, which decrypts the cipher text and sends the plain text as part of the REST API response.

System and Access Requirements

- Oracle Integration, Version 22.1.3.0.0 or higher

- OCI Vault
- An account on OCI with the Administrator role

Before You Install the Recipe

To access OCI Vault from Oracle Integration and encrypt or decrypt data, you'll require to perform certain configuration tasks on OCI.

Log in to your OCI instance as an **Administrator** and perform the following tasks.

1. Create an OCI Vault for your master encryption keys. See Managing Vaults.
2. Create a new master encryption key. See Managing Keys.

> **Note:**
>
> You can also upload your own master encryption keys if you want to manage your encryption keys.

3. Get your Tenancy and User OCIDs. See Where to Get the Tenancy's OCID and User's OCID.
4. Create your own private key or generate it from the OCI console. See How to Generate an API Signing Key.
5. Get the key's fingerprint. See How to Get the Key's Fingerprint.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources.

Configure the Oracle OCI Vault Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave **REST API Base URL** selected.
Connection URL	Enter the Cryptographic Endpoint URL of the Vault you created earlier. See Before You Install the Recipe.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Select OCI Signature Version 1**.
Tenancy OCID	Enter the Tenancy OCID obtained earlier. See Before You Install the Recipe.
User OCID	Enter the User OCID obtained earlier.

Field	Information to Enter
Private Key	Enter the API key generated earlier. **Note:** Before you upload the private key, you must convert it into the PKCS1 format.
Finger Print	Enter the key's fingerprint obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Lookup Table

Edit the **Oracle_Encrypt_Decrypt_Data** lookup table to update the configurable values.

1. In the Lookups section, click the lookup name.
2. Enter the appropriate values for the following keys.

Key	Description
KeyId	Enter the OCID of the master encryption key created earlier. See Before You Install the Recipe.
EmailTo	Enter an email address to which run-time exception emails are sent.
EmailFrom	Enter an email address from which run-time exception emails are sent.
ApiVersion	Enter the correct REST API version of OCI Vault REST APIs. To obtain the API version, see API Reference and Endpoints.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. In the project workspace, click **Activate**. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.

   A message confirms that the integration has been activated. Refresh the page to view the updated status of the integration.
2. Run the recipe from an external application.
   a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

b. On the Configure and run page, click **Endpoint metadata**.

c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.

d. From the external application, send a `POST` request to this endpoint URL along with the plain text or cipher text, which you want to encrypt or decrypt. Provide the text in the `POST` request's **Body** field. See the subsequent step for example request payloads.

The recipe encrypts or decrypts the data you pass in the request body.

3. Test the recipe in Oracle Integration.

a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

b. On the Configure and run page, in the Request section, click **Body** and enter the request data.

Example request payload to encrypt data:

```
{
      "plain_text": "oracle"

}
```

Example request payload to decrypt data:

```
{
     "cipher_text":
"QW3HfP0AD4bvAFJgNV2RrrLrbuNAmKxa2MhaUhFlsgUU2DwAgMWxp6MWZFcAAAAA"

}
```

c. Click **Run**

The recipe invokes an OCI Vault REST API to encrypt or decrypt the data. The Activity Stream pane appears displaying the status of the integration instance's execution.

d. In the Response section of the Configure and run page, under the **Body** tab, you'll find a success response, `200 OK`.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Oracle Cloud Infrastructure Documentation*

Export Employee Images from Oracle HCM Cloud to an FTP Server

Use this recipe to export employee images from Oracle HCM Cloud to an FTP server.

Topics:

- Activate and Run the Recipe

About This Recipe

This recipe extracts employee images from Oracle HCM Cloud and exports them to an FTP server.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When an employee image is added or updated in Oracle HCM Cloud, the recipe exports it to an FTP server. Further, when an employee image is deleted from Oracle HCM Cloud, the recipe generates a CSV file with the employee number and ID.

When the integration flow of the recipe is triggered, the following happens:

- The integration flow calls an Oracle HCM custom Enterprise Scheduler (ESS) job using the export bulk data option of the Oracle ERP Cloud adapter.
- The custom ESS job extracts employee image changes incrementally from Oracle HCM Cloud. That is, it extracts all records from last successful run of the ESS job. Subsequently, the ESS job exports the employee images as XML files to an FTP server.
- Once ESS job's execution is complete, a callback is sent to Oracle Integration.
- The integration flow reads the XML files placed in the FTP server by the ESS job, processes the data, and sends back the images (JPG files) to the FTP server. In case of deleted image files, a CSV file is generated with employee number and ID.
- If an error is detected, a notification email is sent to the email Id specified in the lookup table.

System and Access Requirements

- Oracle Integration, Version 24.02 or higher
- Oracle Fusion SaaS
- An account on Oracle Fusion SaaS with the Administrator role
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Fusion SaaS instance, Oracle ERP Cloud, and FTP server in order to successfully connect to these external systems using Oracle Integration and export employee images.

Access Your FTP Server

Obtain an FTP server and ensure that you can access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server to add the XML data files and employee images. Note down the path of the directory.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see Assign Required Roles to an Integration User.
   - `Integration Specialist`
   - `Oracle ERP Cloud-specific data access to the integration user`

Configure Oracle Fusion SaaS

Complete the following tasks in your Oracle Fusion SaaS instance to successfully connect to it from Oracle Integration and export employee images.

1. Add an FTP server or sFTP server as a delivery channel from the Administration page of Oracle Business Intelligence. See Add an FTP or SFTP Server.
2. Enable auditing on **Person Image**. See Auditing Community Development Data.
3. Deploy the custom Business Intelligence Publisher (BIP) Data Model and Business Intelligence Publisher Report. You can download the `EmployeeImageExtract.catalog` file to use in the recipe from BI Publisher catalog.
4. Update **PARAMETER4** of bursting code in BIP Data Model with the folder path that you created in your FTP server. See Access Your FTP Server.
5. Deploy the custom ESS job. See Creating an Oracle Enterprise Scheduler (ESS) Job.

   Specify the following values for the fields while creating the custom ESS job.

Field	Information to Enter
Display Name	Oracle HCM FTP Extract Employee Images
Name	ExtractEmployeeImages
Path	/ExtractEmployeeImages/
Job Application Name	EarHcmEss
Job Type	BIP job type
Enable Bursting Report	Set this property to enable bursting report.
Report ID	/Custom/Human Capital Management/EmployeeImageExtract/EmployeeImageExtractReport.xdo

> **Note:**
>
> If you want to customize any of the above-mentioned values, ensure that you make necessary changes in the lookup table and BIP Data Model. Deploy the BIP Report in the path specified in Report ID.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection (Trigger, Invoke)

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the user name of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud.
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Oracle FTP Connection (Trigger, Invoke)

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Name	Enter the host address of your sFTP server.
FTP Server Port	22.
SFTP Connection	Select **Yes** from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **FTP Server Access Policy**.
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**  .

Configure the Lookup Table

Edit the **Oracle HCM FTP Extract Employee Images** lookup table and update the lookup keys with appropriate values.

1. In the Lookups section, click the lookup name.
2. In the lookup table, specify values for the following keys.

Key	Description
IntegrationID	The key to fetch all the remaining values from the lookup. For example: `ORCL-HCM_FTP_EXTRACT_EMP_IMG`
EssJobName	The name of the ESS job defined in the Oracle Fusion SaaS instance. For example: `ExtractEmployeeImages`
EssJobPath	The ESS job path defined in Oracle Fusion SaaS instance. For example: `/company_name/apps/ess/custom/ExtractEmployeeImages/`
OICSFTPIn	The FTP location where the ESS job output files must be stored. Enter the target location's path. For example: `/home/users/In`
OICSFTPOut	The FTP location where the images and CSV files must be stored. Enter the target location's path. For example: `/home/users/Out`
OICSFTPArchive	The FTP location where the processed ESS job task files must be stored. Enter the target location's path. For example: `/home/users/Archive`
OICSFTPError	The FTP location where the error files must be stored. Enter the target location's path. For example: `/home/users/Error`
FilePrefix	The file prefix of the ESS job outputs. For example: `IMAGE_*.*`
FromEmailAddress	Specifies the email address to which the error notifications are sent. Enter an email address of your choice.
ToEmailAddress	Specifies the email address from which the error notifications are sent. Enter an email address of your choice.
EmailSubject	Specifies the subject of the email error notification. For example: `Export Employee Images Status`

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Log in to your FTP server and check if the employee image files have been exported to the output directory.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Export Employee Records from Workday to Azure Active Directory

Use this recipe to export employee records from Workday to Azure Active Directory.

Topics:

About This Recipe

This recipe exports employee records from Workday to Azure Active Directory on a scheduled basis. It creates user records in Azure Active Directory according to the employee data received from Workday.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow

reads the employee records from the Workday instance and exports them to Azure Active Directory.

While running the recipe, you can configure the integration flow to read Workday employee records created after a specified time, thereby picking up only the new or required employee records in each run. Basic data such as name, ID, date of birth, email, and so on are synchronized between Workday and Azure Active Directory.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Workday
- An account on Workday with the Administrator role
- Azure Active Directory
- An account on Azure Active Directory with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Workday and Azure Active Directory instances to successfully connect to these external systems using Oracle Integration and export employee records.

Configure Workday

To access Workday using Oracle Integration and export employee records, you must perform certain configuration tasks on your Workday instance.

For general configuration tasks, see Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.

In addition, you must create an Enterprise Interface Builder (EIB) report to export Workday data in bulk. And, before you create an EIB report, you must create a custom report with custom fields, calculated fields, and business objects.

1. Create calculated fields.
   a. In the quick find box, search for **Create Calculated Fields** task.
   b. In the pop-up window, enter the **Field name**. For example: `CF_Most Recent Hire Event - Date and Time Completed`. Also, select the **Business Object** as **Worker**, and **Function** as **Lookup Related Value**.
   c. Click **OK**. You are redirected to Create Calculated Field - Lookup Related Value page.
   d. In the **Condition** column, create the following calculated fields:

Calculated Field	Details
CF_Most Recent Hire Event - Date and Time Completed	**Business Object**: Worker **Lookup Field**: CF_Most Recent Hire Event (CW or EE) **Related Business Object**: Staffing Event **Return Value**: Date and Time Completed **Function**: Lookup Related Value

Calculated Field	Details
CF_Most Recent Hire Event (CW or EE)	**Business Object**: Worker **Source Field**: Staffing History - Approved **Related Business Object**: Staffing Event **Condition**: Is Hire Transaction **Sort Field**: Effective Date **Sort Direction**: Descending (Z to A) **Instance to be Returned**: First occurrence **Function**: Lookup Related Value

Configuration details for other related calculated fields are listed below.

Calculated Field	Details
CF_Hire Event	**Business Object**: Action Event **Lookup Field**: Worker **Related Business Object**: Worker **Return Value**: CF_Most Recent Hire Event (CW or EE) **Function**: Lookup Related Value
CF_Hire Event	**Business Object**: Worker **Lookup Field**: Worker **Related Business Object**: Worker **Return Value**: CF_Most Recent Hire Event (CW or EE) **Function**: Lookup Related Value

2. Create a custom report.
   a. Log in to the Workday instance.
   b. In the quick find box, search for **Create custom report**.
   c. In the Create Custom Report window:
      i. Enter the **Report Name**. For example: `RPT_INT_SOA_Test_11`.
      ii. Select the **Report Type** as **Advanced**.
      iii. Select the **Enable As Web Service** check box.
      iv. Deselect the **Optimized for Performance** check box.
      v. Select the **Data Source** as **All Workers**.
      vi. Click **OK**. You are redirected to Edit Custom Report page.
      vii. Under the **Columns** tab, add the **Fields** and **Custom Fields** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

Business Object	Field	Column Heading Override	Column Heading Override XML Alias
Worker	First Name	First Name	First_Name
Worker	Last Name		lastName
Worker	Email - Work	Emails	Email_-_Work

Business Object	Field	Column Heading Override	Column Heading Override XML Alias
Worker	Employee ID		Employee_ID
Worker	Worker Status		Worker_Status
Worker	Date of Birth		dateOfBirth
Worker	Supervisory Organization		Department
Worker	Work Address - City	Primary Address - City	Primary_Address
Worker	Work Address - Country	Primary Address - Country Name	Phone_Work
Worker	Work Address - Formatted Line 1	Primary Address - Formatted Line 1	Company_-_Name
Worker	Work Address - Formatted Line 2	Primary Address - Formatted Line 2	Primary_Address_-_Formatted_Line_2
Worker	Phone - Primary Work	Phone - Primary Work	primaryWorkPhone
Worker	Business Title		Business_Title

d. Click **OK**.

e. After creating custom reports, you can specify how the report results should be sorted.

f. Under the **Sort** tab, add the **Fields** and **Sort Direction** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

Field	Sort Direction
CF_Most Recent Hire Event - Date and Time Completed	Alphabetical - Ascending

g. Under the **Filter** tab, add the **Fields** and **Custom Fields** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

And/Or	(	Field	Operator	Comparison Type	Comparison Value	)	Indexed
And		Last Name	is not blank				
And		Email - Work	is not empty				
And		Hire Date	greater than or equal to	Prompt the user for the value	Starting Prompt		
And		Hire Date	less than or equal to	Prompt the user for the value	Ending Prompt		

h. In the Prompts section, select the **Populate Undefined Prompt Defaults** check box.

> **Note:**
>
> The data source **All Workers** uses built-in prompts such as **Start Date**, **End Date**, and so on. Based on the report you have chosen, Workday prompts you to fill in specific reporting criteria.

3. Create an EIB report.
   a. Log in to the Workday instance.
   b. In the quick find box, search for **Create EIB** task.
   c. Enter the **Name** of the EIB task (`RPT_INT_SOA_Test_11`).
   d. In the Create EIB pop-up, choose **Outbound** and click **OK**.
   e. In the General Settings, enter comments and then click **Next**.
   f. Under the **Get Data** tab, select the **Data Source Type** as **Custom Report**.
   g. Select the **Custom Report** as **RPT_INT_SOA_Test_11** and then click **Next**.
   h. Under the **Transform** tab, select the **Transformation Type** as **New Custom Report Transformation**, then click **Next**.
   i. Under the **Deliver** tab, perform the following steps.
      i. Select the **Delivery Method** as **Workday Attachment**.
      ii. Enter the **File Name** with the relevant file extension. For example, `output.csv`.
      iii. Specify the **Document Retention Policy (in Days)** as **should be less than 180**, and then click **Next**.
      iv. In the Details section, select the **MIME Type** as **CSV**.
      v. Click **Next**.
   j. Under the **Review and Submit** tab, verify the details, then click **OK**.
   k. Once you create the EIB report, the View Integration System window appears. In the top-left corner, click the ellipsis next to the EIB task name, choose **Enterprise Interface**, and then select **Configure Transformation**.
   l. In the Configure Transformation window, update the report transformation details, click **OK**, and then click **Done**.
   m. In the View Integration System window, in the top-left corner, click the ellipsis next to the EIB task name, choose **Integration**, and then select **Launch/Schedule**.
   n. In the Launch/Schedule Integration pop-up, select the **Run Frequency** as **Run Now**, and then click **OK**.
   o. In the Schedule an Integration window, specify the **Start Date** and **Hire Date**, and then click **OK**.

   The EIB report gets generated.

Configure Azure Active Directory

To configure Azure Active Directory, see Prerequisites for Creating a Connection in *Using the Azure Active Directory Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Workday Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the Workday host name. For example: `https://wd2-impl-services1.workday.com`.
Tenant Name	Enter the tenant name of the account created for Oracle Integration on Workday. See Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Workday Username Token Policy**.
Workday Integration User	Enter the username of the account created for Oracle Integration on Workday. See Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.
Password	Enter the password of the account created for Oracle Integration on Workday.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Azure AD Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the tenant ID in the **Tenant ID** field.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Client Credentials**.
Client ID	Enter the client ID. See Configure Azure Active Directory.
Client Secret	Enter the client password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Lookup Tables

The recipe contains the following lookup tables. Edit them as necessary.

- **ORACLE-BRT_LOOP_LOOKUP**: Use this lookup to configure the loop count.

Property	Description	Value
MaxLoopCount	Specify the value less than or equal 10	For example: 10

- **ORACLE-BRT_AZURE_AD_LOOKUP**: Use this lookup to configure user account enabling, the domain associated with the user principal name (UPN) used for identification, and password.

Property	Description	Value
AccountEnabled	Specify the value as either True or False.	true
UserPrincipalNameDomain	Stores the User Principal name domain.	@abc.xyz.com
PasswordSignIN	Specify the value as either True or False.	true

- **ORACLE-BRT_WORKDAY_LOOKUP**: Use this lookup to configure Integration System Reference ID and Type.

Property	Description	Value
IntegrationSystemID	Specify the Integration System Reference ID.	INT_SOA_Test_11_CSV
IntegrationSystemType	Specify the Integration System Reference Type.	Integration_System_ID

1. In the Lookups section, click the lookup name.
2. In the **Value** column, enter or map the appropriate values for the properties.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Update property values for the **Oracle Workday Azure AD UserSync** integration flow. You can update the integration properties only after activating this flow.

a. In the Integrations section, click **Actions** ⋮ on the integration flow, then select **Update property values**.

b. In the Update Property Values dialog, update the following:

Integration Property	Description
EmailTo	Specify the email address to which run-time exception emails are sent. Enter an email address of your choice.
RetryCount	Number of retries when an invoke fails.
WaitForAzureADRetry	Time in seconds for retrying user creation inAzure Active Directory.
WaitForEIB	The wait time in seconds specified for an EIB report to be retrieved.

c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

3. Run the integration **Oracle Workday Azure AD UserSync** to create users in Azure Active Directory according to the employee data received from Workday.

a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

The Configure and run page is displayed with the **endDate** and **startDate** parameters. Specify a current value for **endDate** and **startDate** in the format yyyy-mm-dd. This will enable the retrieval of details for employees who were hired within that specified date range.

b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The recipe now creates users in Azure Active Directory corresponding to the employee data exported from Workday.

> **Note:**
>
> You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

4. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.

5. Log in to Azure Active Directory and check if users are created for employee records exported from Workday.

View the employee records.

a. In the **Filter** box, search for **Users**.

b. Under **Users**, select **All Users**.

The list of users created is displayed.

Related Documentation

- *Using the Workday Adapter with Oracle Integration 3*
- *Using the Azure Active Directory Adapter with Oracle Integration 3*

Handle Throttling Using Parking Lot Pattern

Use this recipe to implement parking lot pattern in order to handle downstream throttling.

Topics:

About This Recipe

This recipe demonstrates how to use the parking lot pattern for handling downstream throttling and errors with an example use case of importing employee bank details from an external system to a downstream payroll system. An Oracle ATP database table is used for implementing the parking lot pattern.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. In this example, employee bank details are received from an external system such as Oracle HCM Cloud in a CSV file. The CSV file is then downloaded to a local directory on FTP Server or File Server. Subsequently, the employees bank details are registered in batches to an Oracle ATP database table and parked there for a specific time period before further processing. Thus, the integration flow has a chance to throttle the number of batches processed concurrently. The ready batches are picked up in a staged manner for processing and the processed data is finally updated to a downstream payroll system.

Overview of the recipe's integration flows:

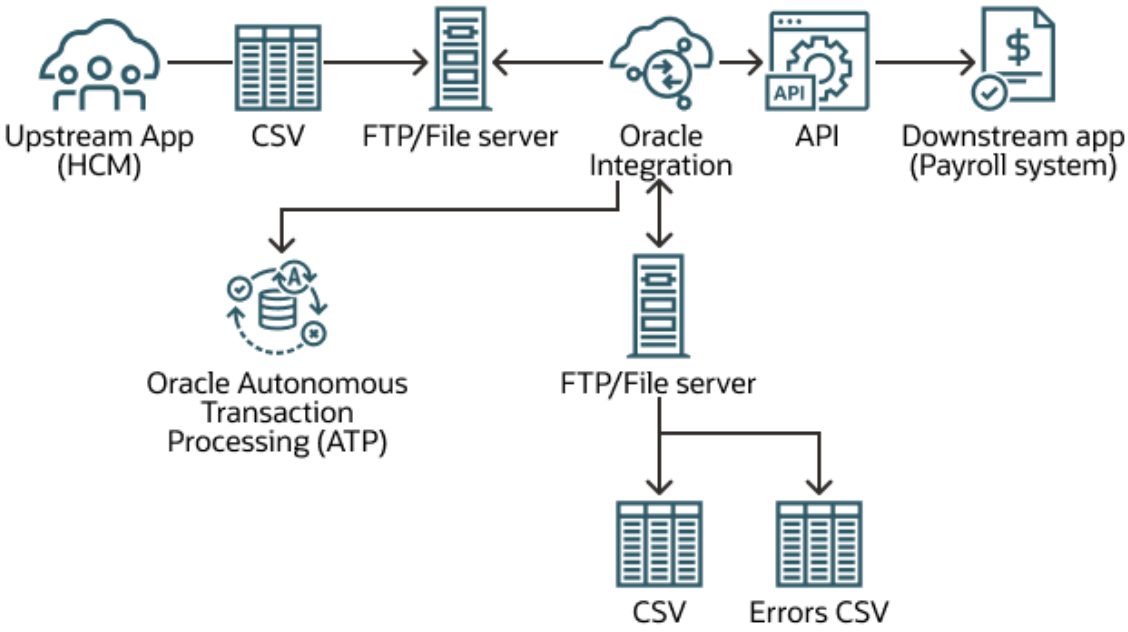
- The **Request Persister** integration (**SR_BulkDownload_RequestPersister_ATP**) reads the CSV file received from external system and writes it to a local input directory in FTP. It reads the batch configurations (such as, group_id and group_type) and writes the metadata to the Oracle ATP database table. The registered batches are parked in the Parking Lot table in ATP for a specific period of time.
- The **Scheduled Dispatcher** integration (**SR_ScheduledDispatcher_CSVBatch**) is scheduled to run at required frequency. At every run, it fetches the batch requests from the Parking Lot table in ATP and dispatches them to the **Async Batch Processor** integration for processing. Thus, it triggers the **Async Bach Processor** integration to process the batches that are submitted for processing.
- The **Async Processor** integration (**SR_OneWay_Processor_HCM_To_Payroll**) processes all the submitted batch requests. Also, it updates the batch status (for example, if a batch is successfully processed, it updates the status to PROCESSED) in the Batch Statistics table in ATP. On successful processing, it invokes another integration (for example, **SR_MOCK_EmpBankAccountProvider** integration in this recipe) to send the processed information to a downstream application via REST API calls. Any errors received in the response from the downstream application are written into an Error folder in FTP. These payload errors are also written to a Payload Error Record table in ATP.
- The **Resubmission Processor** integration (**SR_ScheduledDispatcher_PayrollErrors**) is used to resubmit any batch requests failures/errors that were fixed. It fetches any record from the Payload Error Record table that are in READY state and re-submits them for processing.

System and Access Requirements

- Oracle Integration 3
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server
- Oracle ATP
- An account on Oracle ATP with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.



A CSV file with employee bank details from an upstream application such as Oracle HCM Cloud is downloaded to an FTP Server or File Server. The first integration flow (**Request Persister**) in Oracle Integration registers the batches in the ATP database. Each batch request is parked in the ATP database parking lot table for a certain period of time so the integration flow has a chance to throttle the number of batches processed concurrently. The second scheduled integration flow (**Scheduled Dispatcher**) in Oracle Integration fetches the batches from the parking lot table in ATP at a date and time of your choice, and dispatches the batches for further processing. The second integration triggers the third integration flow (**Async Processor**) which processes the dispatched batches. The processed batches are then sent to a downstream application such as a Payroll system via a REST API call. Any errors that are received as response from the downstream application are rectified and then resubmitted for processing by the fourth integration flow (**Resubmission Processor**) in Oracle Integration.

Before You Install the Recipe

You must perform the following configuration tasks on your FTP Server and Oracle ATP instance in order to connect to these external systems using Oracle Integration and successfully handle throttling with the implementation of parking lot pattern.

Access Your FTP Server

Obtain a sFTP server and ensure that you're able to access it.

1. Log in to the FTP server using your user name and password through an FTP client; for example, FileZilla.
2. Create the following directories:
   - An input directory to place the CSV file received from the upstream system. For example: `HCM_To_Payroll_Input`. This file should be in a particular format for the recipe to run successfully. Download the following sample input file to successfully test the recipe: Sample file.
   - A directory to place the files from the upstream system that were successfully processed. For example: `HCM_Archive`
   - A directory to receive files from the downstream system that could not be processed successfully. For example: `Payroll_To HCM_Input`
   - A directory to archive files received from the downstream stream that were retried and processed successfully. For example: `Payroll_Archive`
   - A local input directory. For example: `Input`
   - A local output directory. For example: `Output`
   - A local error directory. For example: `Error`
   - A local processing directory. For example: `Processing`

You can also use File Server, an embedded sFTP server within Oracle Integration to store and work with the files. To use File Server, you must enable it for your Oracle Integration instance. See Enable File Server in *Using File Server in Oracle Integration 3*.

Configure Oracle ATP

To configure Oracle ATP, perform the following configuration tasks:

1. Complete the prerequisites for creating a connection from Oracle Integration to Oracle ATP. See Prerequisites for Creating a Connection in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.
2. Download and run the following SQL script to create the required parking lot table: SQL Script.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources.

Configure the Oracle FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Name	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select **Yes** from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **FTP Server Access Policy**.
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Oracle ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See Configure Oracle ATP.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **JDBC With OCI Signature**.
Wallet	Select the check box, then click **Upload** to upload the wallet file. See Configure Oracle ATP.
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.
Object Storage Region	Specify the region in which your object storage is located. For example, `us-ashburn-1`.

Field	Information to Enter
Object Storage Tenancy OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, `ocid1.tenancy.oc1.alphanumeric.value`
Object Storage Compartment OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, `ocid1.compartment.oc1.alphanumeric.value`
Object Storage User OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, `ocid1.user.oc1.alphanumeric.value`
Private Key	Click **Upload** to select the key you created. Ensure that the key is in RSA (PKCS1) format. **Note:** Only a private key *without* a pass phrase/ password is supported.
Finger Print	Enter the finger print that was generated when you created the key in the Oracle Cloud Infrastructure Console.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

The recipe contains the following lookup tables that you can edit and configure as per your requirement.

- **Email_Notification**
- **Group_Setup**

1. In the Lookups section, click the lookup name.
2. Edit the following lookup tables.
   a. In the **Email_Notification** lookup table, edit the value of the **BatchNotificationTo** lookup key by entering an email address of your choice to which notification error emails have to be sent.
   b. In the **Group_Setup** lookup table, use the default directories provided for **GRPID: HCM_To_Payroll** or edit them as required with the directories that you created in your FTP server. See Access Your FTP Server.
      - **InputDir**: Enter the directory path to receive and store the input files from the upstream system (for example: `/pob-Workshop/HCM_To_Payroll_Input`).
      - **ArchiveDir**: Enter the directory path to archive files from the upstream system which were successfully processed (for example: `/pob-Workshop/HCM_Archive`.
      - **ProcessingHomeDir**: Enter the path of the home directory (for example: `/batchHome`).

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator
2. Run the recipe.

   Ensure that a sample input file with employee bank details in the correct format is uploaded to the input folder (in this case, `HCM_To_Payroll_Input`).

   a. Run the **SR_BulkDownload_RequestPersister_ATP** integration flow.

      i. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

      ii. On the Configure and run page, click **Run**.

   You've now successfully submitted the integration for execution. The integration now writes the input file to a local input folder (`/Input`). The information from the file, that is, the file metadata is persisted to the parking lot table (PAYLOAD_PARKING_LOT_TAB) in Oracle ATP.

   > **Note:**
   >
   > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

   b. Run the **SR_ScheduledDispatcher_CSVBatch** integration flow.

      i. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

      ii. On the Configure and run page, enter a value in the throttling parameter **MaxRecords_fromDB** to specify the number of records that should be fetched per run from the parking lot table (PAYLOAD_PARKING_LOT_TAB) in Oracle ATP.

      iii. On the Configure and run page, click **Run**.

   You've now successfully submitted the integration for execution. The integration now reads the data from the parking lot table, and dispatches the data to the asynchronous integration flow **SR_OneWay_Processor_HCM_To_Payroll**, thus triggering the integration flow to process the batch records fetched from the parking lot table (PAYLOAD_PARKING_LOT_TAB). In the parking lot table, the status of the batch records that were submitted for processing changes from NEW to PROCESSED.

   > **Note:**
   >
   > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

The **SR_OneWay_Processor_HCM_To_Payroll** integration flow processes the batch files and calls the downstream application via a REST API call. It updates the batch statistics (BATCH_STATISTICS) table in ATP with the status of each record that is being processed.

The records that the downstream application could not process are written to the payload error table (PAYLOAD_ERRORS_TAB) in ATP with the status as ERRORED. When the errors are rectified, the status of the records changes to READY.

The **SR_ScheduledDispatcher_PayrollErrors** integration flow can be run to fetch the READY records from the payload error table (PAYLOAD_ERRORS_TAB), process them, and send them to the downstream application.

3. Monitor the running of the integration flows in Oracle Integration. See Monitor Integrations

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Import a File into Oracle Content Management

This recipe imports a file into Oracle Content Management.

Topics:

About This Recipe

This recipe receives a file as input and sends the file as an attachment to a specific folder in Oracle Content Management.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. The **Oracle REST Content Management Create Folder** integration receives folder name as input and creates a folder in Oracle Content Management. Subsequently, the **Oracle REST Content Management Upload File** integration receives file name, file content, and parent folder ID as input and imports the received file into the folder created earlier.

> **Note:**
>
> You can only create one folder and import one file with each trigger.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Oracle Content Management
- An account on Oracle Content Management with the Administrator role

Before You Install the Recipe

To access Oracle Content Management using Oracle Integration and import a file, you must perform certain configuration tasks on your Oracle Content Management instance.

1. Create a parent folder in Oracle Content Management in which the new folder must be created. See **Create a new folder** in Work with Your Files and Folders.

   Note the `folderName`, `parentId`, and `description` of the parent folder.
2. Assign **Role for new public links** as **Contributor**. See **Change folder sharing settings** in Work with Your Folders and Files.
3. Add a member to the folder. See **Add Folder Members** in Share Folders.
4. Create an OAuth client and acquire the client ID and secret. See Create an OAuth Client and Acquire a Client ID and Secret.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Trigger Connection

Use the following steps to configure the **ORCL-BRT-REST-ERP-CONNECTION** connection.

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
3. To return to the project workspace, click **Go back**  .

Configure the Oracle REST Content Management Connection

Use the following steps to configure the **ORCL-BRT-CM-CONNECTION** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select **REST API Base URL**.
Connection URL	Enter your Oracle Integration instance URL. For instance, `https://<instance-name>.com/`.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **OAuth Client Credentials**.

Field	Information to Enter
Access Token URI	Enter the access token URL of Oracle Content Management.
Client Id	Enter the client ID of the OAuth application created in Oracle Content Management. See Before You Install the Recipe.
Client Secret	Enter the client secret of the OAuth application created in Oracle Content Management.
Scope	Enter the scope of the OAuth application created in Oracle Content Management.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe from an external application to create a folder.
   a. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle REST Content Management Create Folder** integration flow, then select **Run**.
   b. On the Configure and run page, click **Endpoint metadata**.
   c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
   d. From the external application, send a `GET` request to the endpoint URL along with parameters such as `folderName`, `parentId`, and `description`. See Before You Install the Recipe.

   The recipe creates a folder in the parent folder in Oracle Content Management.
3. Run the recipe from an external application to import a file into the parent folder.
   a. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle REST Content Management Upload File** integration flow, then select **Run**.
   b. On the Configure and run page, click **Endpoint metadata**.
   c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
   d. From the external application, send a `POST` request to this endpoint URL along with the file details. Provide the text in the `POST` request's **Body** field.

      Example request payload:

```
{
  "contentype": "application/json",
  "filename": "<File name; For instance: Test.txt>",
  "FolderId": "<Parent ID from the earlier response>",
```

```
  "FileContent": "<Enter file content in Base64 decode format>"
}
```

The recipe creates the file with the file name and content you pass in the request body.

4. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
5. Log in to the Oracle Content Management and check for the new file and folder.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Import a Payment Request into Oracle ERP Cloud

Use this recipe to import a payment request from an FTP server to Oracle ERP Cloud.

Topics:

About This Recipe

This recipe imports an interface file containing payment requests from an FTP server to Oracle ERP Cloud according to a schedule specified in Oracle Integration.

In Oracle ERP Cloud, a payment request is used to make a one-time payment (without creating a supplier record) to external suppliers or parties. When a payment request (transaction) is imported into Oracle ERP Cloud, the payees, parties, and bank accounts are automatically created in the application, thereby simplifying the task of making the payment. In addition, the transactions are validated automatically, eliminating the need for manual intervention.

As part of this recipe, Oracle provides an example file structure that you can use to load a payment request into Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the recipe manually or specify an execution schedule for it. When triggered, the recipe loads a payment request from a specified FTP server to Oracle ERP Cloud.

System and Access Requirements

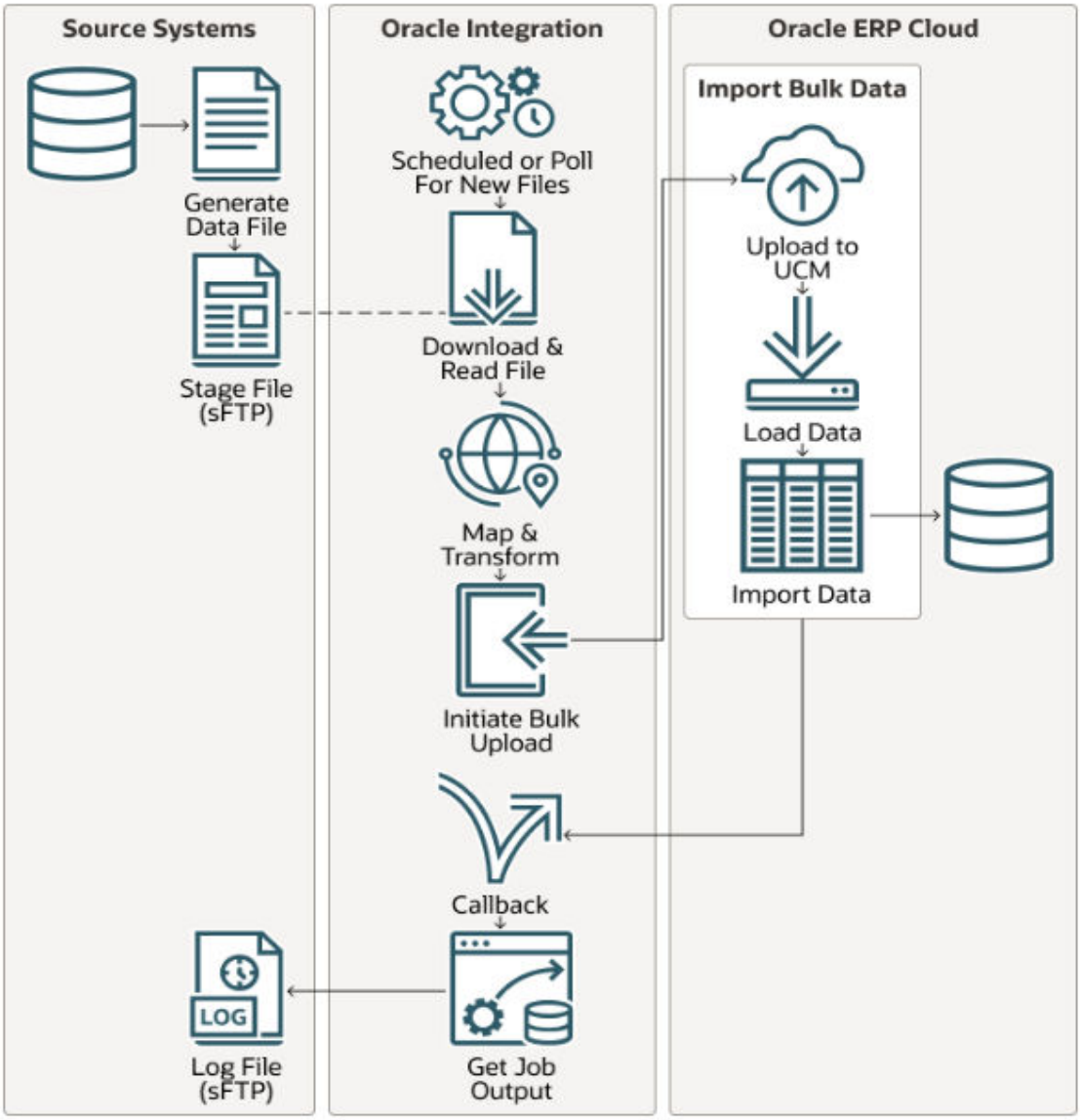
- Oracle Integration, Version 21.4.2.0.0 or higher
- Oracle ERP Cloud, Version 21d or higher
- Accounts on Oracle ERP Cloud with the Administrator role and the Accounts Payable Invoice Supervisor/Accounts Payable Manager/Accounts Payable Specialist role (in order to check for payment requests)
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server.

Recipe Schema

This section provides an architectural overview of the recipe.

The primary integration flow (**Oracle FTP ERP Cloud Payment Request**), when triggered by an execution schedule or a manual submission, queries the FTP server for a payment request file (`.zip`). If a file is found, the integration flow processes the file. Initially, the flow maps the file's layout to the layout required by Oracle ERP Cloud, and then it imports the payment requests in bulk through the ERP Adapter.

After importing the payment request, Oracle ERP Cloud makes a callback to the secondary integration flow (**Oracle FTP ERP Cloud Payment Request Callback**), which downloads the log files from Oracle ERP Cloud and places them on the FTP server.



Payment Request File Definition

This section describes the file layout used by the primary integration flow (**Oracle FTP ERP Cloud Payment Request**) of the recipe. This integration flow is configured to look for a `.zip` file on your sFTP server. The `.zip` file must contain the following two files:

- `apPaymentRequest.properties`
- `apPaymentRequest.csv`

Definition: apPaymentRequest.properties

This file tells Oracle ERP Cloud how to process and validate the payment requests. This file is in the `.csv` format.

Parameter	Value
Program Location	`oracle/ apps/ess/financials/payables/invoices/ transactions/`
Program	`APXPRIMPT`
Interface	The file name to be loaded, this will be the the name placed within the FTPImportFileName lookup without the `.zip` suffix. For example: `apPaymentRequest`.
Business Unit	Setup and Maintenance/Manage Business Unit.
Accounting Date	Enter an accounting date. If you don't enter a date, the Accounting Date is based on the Accounting Date Basis option.
Oracle_Int1	Hard-coded internal value set to `1000`.
Source	For Payment Requests, this is always `ONE_TIME_PAYMENTS`.
Import Set	Name of an import set to limit the import.
Purge	Yes, for Payables to delete records that match the Source and Group ID of this import. No, to keep all the records except the rejected records.
Summarise	Enter whether to summarize the audit portion of the Payables Payment Request Interface Import Report.
Ledger	Setup and Maintenance/Manage Primary Ledgers.
Invoice Group	Enter the name of an invoice group.
Oracle_Int2	Hard-coded internal value set to 1.

Example Values: apPaymentRequest.properties

The following table provides an example set of values for the parameter file. Ensure that you enter the values specific to your configuration in the Interface, Business Unit, and Ledger fields.

Parameter	Value
Program Location	`oracle/apps/ess/financials/payables/invoices/ transactions/`
Program	`APXPRIMPT`
Interface	`apPaymentRequest`
Business Unit	`300000047498175`

Parameter	Value
Accounting Date	null
Oracle_Int1	1000
Source	ONE_TIME_PAYMENTS
Import Set	null
Purge	N
Summarise	N
Ledger	300000047488112
Invoice Group	null
Oracle_Int2	1

Example File: apPaymentRequest.properties

This section provides an example parameter file, with the name: `apPaymentRequest.properties`.

The file name must be same as the value entered in the Interface field within the file, with `.properties` as suffix.

Example File:

```
ProgramLocation,Program,Interface,BusinessUnit,AccountingDate,Oracle_Int1,Source,
ImportSet,Purge,Summarise,Ledger,InvoiceGroup,Oracle_Int2 /oracle/apps/ess/
financials/payables/invoices/
transactions/,APXPRIMPT,apPaymentRequest,300000047498175,null,1000,ONE_TIME_PAYME
NTS,null,N,N,300000047488112,null,1
```

Definition: apPaymentRequest.csv

This file contains the payment requests that are paid through Oracle ERP Cloud. The file provided here is an example, containing the minimum set of data required to load a payment request.

Prepare this file in the .csv format, using the following structure:

Parameter	Description
Transaction Identifier	Identifier that uniquely identifies a payment request line record.
Party Name	Name of the party to whom the payment is to be made.
Party Original System Reference	Party reference from the original system or external source.
Party Type	Type of party receiving the payment.
Location Original System Reference	Original system reference for the party site address.
Country	Location of the address by country.
Address Line 1	First line of the party address.
Address Line 2	Second line of the party address.
City	City of party to whom the payment is to be made.
State	State of party to whom the payment is to be made.
Province	Province of party to whom the payment is to be made.
County	County of party to whom the payment is to be made.
Postal Code	Postal code of party to whom the payment is to be made.

Parameter	Description
Postal Plus 4 code	Four-digit extension to postal code.
Addressee	Person or recipient located at the address.
Global Location Number	Number that uniquely identifies each global location in a trading-partner enterprise.
Language	Operating language of the party-site address.
Phone Country Code	Country code for the telephone number.
Phone Area Code	Area code for address phone number.
Phone	Phone number of the party receiving payment.
Phone Extension	Phone extension of the party receiving payment.
Payee E-Mail address	E-mail address of the payee to whom the payment is to be made.
Party Tax Registration Number	Unique identifier assigned to a party or party site by a tax authority.
Account Country Code	Country code of the bank account of payee.
Account Currency Code	Currency code of the bank account of payee.
Account Number	Bank account number of the payee.
IBAN	International Bank Account Number of payee.
Secondary Account Reference	Account number reference such as Building Society Roll Number.
Account Suffix	One to three digit number at the end of the account number that determines the kind of account.
Account Name	Name of the payee bank account.
Account Type Code	Type of bank account, such as SAVINGS and CHECKING.
Bank Name	Name of the payee bank to whom the payment is to be made.
Bank Number	Number of the payee bank to whom the payment is to be made.
Bank Branch Name	Name of the branch of the payee bank.
Bank Branch Number	Number of the bank branch. This is known as the routing transit number in US and Canada.
BIC	SWIFT or Bank Identification Code (BIC) of the bank branch.
Check Digits	One or more digits used to validate a bank account number.
Account Description	Description of the payee bank account.
Account Alternate Name	Alternate account name of the bank account of payee.
Business Unit	Sold-to business unit name for which the transaction is being created.
Source	Name of the external system from which a payment request is imported.
Invoice Number	Unique number for payment request.
Invoice Date	Date on which payment request is submitted.
Currency	Currency of the payment request.
Description	Information describing the invoice created for payment.
Import Set	Name or number assigned to the import set containing payment requests.
Legal Entity	The name of the legal entity that belongs to your own corporate structure (enterprise).
Payment Terms	Name of the payment terms used to create installments and to calculate due dates, discount dates, and discount amounts for each invoice.

Parameter	Description
Payment Method	Code identifying the payment method.
Pay Group	Used to group payment requests into a single pay run.
Liability Combination	Account to which liability is recorded in accrual basis of accounting.
Line Number	The number of the invoice line.
Amount	The invoice line amount to import for payment.
Line Description	Line description.
Distribution Combination	Account combination used for creating invoice distribution.

Example Values: apPaymentRequest.csv

The following table provides an example set of values for the payment-request file.

Parameter	Description
Transaction Identifier	100109
Party Name	Fred Jones
Party Original System Reference	OTP-PTY-100109
Party Type	ORGANIZATION
Location Original System Reference	OTP-PTY-100109
Country	GB
Address Line 1	Oracle Parkway
Address Line 2	Thames Valley Park
City	Reading
State	UK
Province	N/A
County	N/A
Postal Code	RG6
Postal Plus 4 code	1RA
Addressee	N/A
Global Location Number	N/A
Language	N/A
Phone Country Code	N/A
Phone Area Code	N/A
Phone	N/A
Phone Extension	N/A
Payee E-Mail address	fred.jones@emaildomain.com
Party Tax Registration Number	N/A
Account Country Code	GB
Account Currency Code	GBP
Account Number	99453336
IBAN	N/A
Secondary Account Reference	N/A
Account Suffix	N/A
Account Name	F Jones

Parameter	Description
Account Type Code	N/A
Bank Name	HSBC
Bank Number	N/A
Bank Branch Name	Reading
Bank Branch Number	754322
BIC	N/A
Check Digits	N/A
Account Description	N/A
Account Alternate Name	N/A
Business Unit	UK Business Unit
Source	ONE_TIME_PAYMENTS
Invoice Number	OTP-FJONES-109
Invoice Date	2021/11/09
Currency	GBP
Description	Refund
Import Set	GBP,Refund,,UK Legal
Legal Entity	UK Legal Entity
Payment Terms	Immediate
Payment Method	N/A
Pay Group	N/A
Liability Combination	303.30.2210.000.000.000
Line Number	1
Amount	20
Line Description	One time payment lines
Distribution Combination	303.30.7740.000.000.000

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration and load payment requests.

Access Your FTP Server and Upload the Files

Obtain an sFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server from which to read the files. Note the path of this directory.
3. Upload a payment-request file (`.zip`) into the directory you've created. You must name the file as `apPaymentRequest.zip`.

> **Note:**
>
> The `.zip` file must contain two files by the name, `apPaymentRequest.properties` and `apPaymentRequest.csv`. For details on the structure of these files, see Payment Request File Definition.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see Assign Required Roles to an Integration User.
   - `Integration Specialist`
   - `AttachmentsUser`
   - `SOAOperator`
   - `FND_MANAGE_CATALOG_SERVICE_PRIV`
   - `Payables Invoice Creation`

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select **Yes** from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **FTP Public Key Authentication.**.

Field	Information to Enter
User Name	Enter the user name to connect to your sFTP server.
Private Key File	Upload the private key file for your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud.
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Lookup Table

Edit the lookup table to specify the FTP server directory from which the file must be read, the name of the file to be read, and an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. In the lookup table, enter the following information.

Property Name	Information to Enter
FTPImportFileName	Specify the name of the file to be read from the FTP server. For example, `apPaymentRequest`. See Access Your FTP Server and Upload the Files.
emailErrorNotificationTo	Enter an email ID of your choice.
FTPImportDirectory	Specify the path of the directory on your FTP server from where the file must be read. For example: `/home/user2/FTP`.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow (**Oracle FTP ERP Cloud Payment Request**, then select **Run**.
   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Check for the new payment request on Oracle ERP Cloud.
   a. Log in to the Oracle ERP Cloud instance using a separate account, for example, as a user with the Accounts Payable Invoice Supervisor/Accounts Payable Manager/ Accounts Payable Specialist role.
   b. Navigate to the Invoice Dashboard screen.
   c. In the **Invoice Number** search field, enter the value of the **Invoice Number** field that's in your interface file, and click **Search**. The imported payment-instruction invoice is displayed.
5. On your FTP server, check the log files from Oracle ERP Cloud.
   a. Log in to the server and navigate to your directory.

      You'll find a file with the following name: `apPaymentsRequest.zip_dateandtimr.zip`.
   b. Download this file from the FTP server to your local system.
   c. Unzip the file and inspect the log files present within.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *File-Based Data Import (FBDI) for Financials*
- *How Payables Payment Request Import Data is Processed*
- *Definition of Available Fields in the Payment Request Integration*

Import Files from File Server to OCI Object Storage

Use this recipe to import files from File Server to OCI Object Storage.

Topics:

About This Recipe

This recipe demonstrates how to import files from File Server to OCI Object Storage using native actions. This solution is connection less, meaning no connections are used to connect to File Server and OCI Object Storage.

- **File Server**: A native action is used to connect to File Server without having to use the FTP adapter connection.
- **OCI Object Storage**: A native action is being used to connect to OCI Object Storage without having to use the REST adapter connection.

To use the recipe, you must install the recipe and configure the resources within it. Subsequently, you can activate and run the integration flow of the recipe. When triggered, the integration flow imports files from File Server into OCI Object Storage.

To use File Server which is an embedded sFTP server within Oracle Integration, you must enable it from yourOracle Integration instance. See Enable File Server in *Using File Server in Oracle Integration 3*.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- File Server
- OCI Object Storage
- An account on OCI Object Storage that has access to create objects

Before You Install the Recipe

You must perform the following configuration tasks on your File Server and OCI Object Storage instances in order to successfully import files from File Server to OCI Object Storage.

Configure File Server

You have to create the required directories in File Server.

1. Create a directory to import the input files from. Note down the path of the directory.
2. Create a directory to archive files that have been processed. Note down the path of the directory.
3. Create a directory where error files are to be placed.

Configure OCI Object Storage

Obtain access to OCI Object Storage and check the following:

1. If you don't already have a bucket where you can upload the imported files, create a bucket in a compartment.
2. Note down the bucket's location.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources.

Configure the Lookup Table

The recipe contains the **SFTPToObjectStorageLKP** lookup table. Edit the lookup keys in the table as per your requirement.

1. In the Lookups section, click the lookup name.
2. Edit the lookup keys in the table.
   - **ToEmailAddress**: Specify the email address to which error notifications have to be sent.
   - **OICErrorDirectory**: Specify the directory in File Server where files that errored out have to be sent.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you have configured the resources of the recipe, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the **FileServer to OCI Storage Native file transfer** integration flow, then select **Run**.

      The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

      - **Bucket Location**: This property holds the location of the bucket where the imported files have to be uploaded in OCI Object Storage.
      - **Bucket Name**: This property holds the name of the OCI Object Storage bucket where the imported files have to be uploaded.
      - **SFTP_File_Path**: This property holds the path on File Server from which files have to be fetched.
      - **SFTP_Archive_File_Path**: This property holds the path on File Server where processed files are archived.

- **File_Name_Pattern**: Specify the file name pattern.

b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

> **Note:**
>
> You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. In OCI Object Storage, check that the files have been imported to the specified bucket. In File Server, check if the processed files have been moved to the archived files directory .

Related Documentation

- Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action
- Interact with Files in File Server

Import Financial Journal Entries from Amazon S3 to Oracle ERP Cloud

Use this recipe to import financial journal entries from an Amazon S3 location to Oracle ERP Cloud.

Topics:

About This Recipe

This recipe reads a file from an Amazon S3 location, does a simple transformation, and then imports the financial journal entries into Oracle ERP Cloud. The recipe uses File-Based Data Integration (FBDI).

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. The integration uses a scheduled orchestration pattern to download the file from the Amazon S3 location, process it, and import it into Oracle ERP Cloud. The import process records the results in a log file. To send this log file back to the Amazon S3 location, the integration uses an app-driven orchestration. The trigger to start an app-driven orchestration is based on an event happening. In this case, the log file is ready for export.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Amazon Web Services (AWS) with Amazon S3

- An account on AWS with Administrator role
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Amazon S3 and Oracle ERP Cloud instances in order to connect to these external systems using Oracle Integration, and successfully import financial journal entries from an Amazon S3 bucket to Oracle ERP Cloud.

Configure Amazon S3

Complete the following configuration tasks on your Amazon S3 instance to connect to it from Oracle Integration and import financial journal entries.

1. Obtain your AWS access keys. See Prerequisites for Creating a Connection in *Using the Amazon Simple Storage Service (S3) Adapter with Oracle Integration 3*.
2. Set up your Amazon S3 bucket where you want the resulting log file to be stored.
   a. Log in to your AWS Management Console at `https://console.aws.amazon.com/`.
   b. In the AWS Management Console, enter `S3` in the search field at the top.
   c. Select **S3 - Scalable Storage in the Cloud** from the search results.

      The Amazon S3 page appears.
   d. Select an existing bucket.
      - Scroll to the Buckets section. All available buckets are listed under it.
      - Select a bucket.
   e. Alternatively, create and choose a new bucket.
      i. Click **Create bucket**.
      ii. In the Create bucket dialog, enter a unique name for the bucket.
      Note some rules about naming an Amazon S3 bucket.
         - A bucket name should be unique across all Amazon S3 buckets.
         - Bucket names must be between 3 and 63 characters long.
         - Bucket names can consist only lowercase letters, numbers, dots (.), and hyphens (-).
         - You cannot write a bucket name as an IP Address, such as 192.168.0.1.
         - Bucket names must begin and end with a letter or number.
         - Bucket names should not contain two adjacent dots (.).
         - Bucket names should not end with `-s3alias`.
         - Bucket names should not start with `xn--`.
   f. In the **AWS Region** field, select a region from the drop-down list.
   g. In the Object Ownership category, leave it as recommended. We use it for controlling the access of the files by specifying roles. If ACLs are disabled, the bucket owner automatically owns and has full control over every object in the bucket.

h. In Block Public Access settings for this bucket category, ensure that **BLOCK ALL PUBLIC ACCESS** is selected. If you want to host your static website in this bucket, you can change the settings later.

i. In the Bucket Versioning category, choose **Disabled**. Bucket versioning is helpful when you want to track any changes in the file made, intentionally or unintentionally. You can see the previous versions of a file, retrieve it, restore it or preserve it.

j. Leave other advanced settings as default.

k. Click **Create bucket**. The new bucket is listed under the Buckets section.

3. Note the AWS Region of the bucket.

   a. Select the bucket.

   b. Click the **Properties** tab, and note the **AWS Region** of the bucket.

Configure Oracle ERP Cloud

Complete the following configuration tasks on your Oracle ERP Cloud instance to connect to it from Oracle Integration, and import financial journal entries.

Perform the general configuration tasks for creating a connection. See Prerequisites for Creating a Connection in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Financial journal entries have to be in a particular format so that Oracle Enterprise Scheduler (ESS) Jobs can be run and the financial data can be finally stored in a log file. To enable this, you have to download the journal import template from Oracle ERP Cloud and prepare a journal import file based on it.

1. Download the journal import template (`JournalimportTemplate.xlsm`).

   a. Navigate to Oracle Fusion Cloud Applications Suite in Oracle Help Center.

   b. Under **Enterprise Resource Planning (ERP)**, click **Financials**.

   c. In the left navigation pane of the Oracle Financials Get Started page, click **All Books**.

   d. Under **Implementation**, open the *File-Based Data Import (FBDI) for Financials* guide.

   e. Under the **General Ledger** section of the guide, click **Journal Import**.

   f. In the **File Links** section, download the journal import template XLSM file (`JournalimportTemplate.xlsm`) and save it to your local machine.

2. Prepare the journal import data file.

   a. Delete the sample data from the `JournalimportTemplate.xlsm` file. Keep only one record for your reference.

   b. Populate the file with a few financial journal entry records.

   c. Then, click and open the Instructions and CSV Generation sheet.

   d. Click **Generate CSV File**.

   e. Save the generated CSV file. It will be saved as a ZIP file on your local machine.

      Upload the data file to the Amazon S3 bucket that you created in your Amazon S3 instance.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle Amazon S3 Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select **REST API Base URL**.
Connection URL	Enter the Amazon S3 connection URL: `https://s3.<your-bucket-region>.amazonaws.com/`

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **AWS Signature Version 4**.
Access Key	Enter the access key that you obtained. See Configure Amazon S3.
Secret Key	Enter the secret key that you obtained.
AWS Region	Select the AWS region where you have created the Amazon S3 bucket.
Service Name	Select **Amazon Simple Storage Service (Amazon S3)**.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud.
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**  .

Configure the Lookup Table

Edit the **ORCL-BRT-S3_IMPORT_JOURNAL_SETTINGS** lookup table with appropriate values.

1. In the Lookups section, click the lookup name.
2. Enter appropriate values in the **Property_value** column for the following properties.

Property_name	Property_value
EmailErrorNotificationTo	Enter the email address where notifications will be sent in case of integration errors.
amazonS3ImportFileName	Enter the name of the financial journal file that have to be imported into Oracle ERP Cloud from the Amazon S3 bucket.
amazonS3BucketName	Enter the name of the Amazon S3 bucket.
amazonS3UploadFileName	Enter the name of the log file that has to be send back to the Amazon S3 location.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Update Integration Properties

The **Oracle AS3 ERP Journal Entry Import Callback** integration flow contains the **JobName** integration property that has to updated with appropriate value.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the **JobName** integration property with appropriate value. The **JobName** integration property holds the name of the Oracle Enterprise Scheduler (ESS) job that you run in Oracle ERP Cloud.
3. Click **Submit**.

   A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. the Integrations section of the project workspace, click **Actions** ... on the **Oracle AS3 ERP Journal Entry Import** integration flow, then select **Run**.
   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution. The **Oracle AS3 ERP Journal Entry Import** integration flow fetches the financial journal entries from

Amazon S3 in the journal import file and imports them to Oracle ERP Cloud. In Oracle ERP Cloud, after the Oracle Enterprise Scheduler (ESS) job runs successfully, the imported financial journal entries are recorded in a log file.

> **Note:**
>
> You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

c. Now, the **Oracle AS3 ERP Journal Entry Import Callback** integration flow gets triggered and it uploads the log file from Oracle ERP Cloud into the specified Amazon S3 bucket.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. In the Amazon S3 instance, navigate to your Amazon S3 bucket and verify if the log file has been uploaded to it.

Related Documentation

- *Using the Amazon Simple Storage Service (S3) Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Import Financial Journal Entries from an FTP server to Oracle ERP Cloud

Use this recipe to import financial journal entries from an FTP Server location to Oracle ERP Cloud.

Topics:

About This Recipe

This recipe reads a file from an FTP location, does a simple transformation, and then imports the financial journal entries into Oracle ERP Cloud. It uses File-Based Data Integration (FBDI). The integration flow can be invoked explicitly either as a web service or as a scheduled orchestration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe.

The integration flow of the recipe performs the following tasks:

- Reads the ledger-entries file from the FTP location.
- Unzip the file.
- Stages the file in Oracle Integration and read each entry.

- Transforms the data as required. For example, this integration changes the value in the currency column.
- Sends the file to Oracle ERP Cloud, configure a callback, and invoke the import process with the required parameters.
- After the import completes, Oracle Integration receives the result of the import from Oracle ERP Cloud. If the import completes successfully, then upload the log file to the FTP server. If the import fails, then send an email notification.

You can store the ledger-entries file on File Server, an embedded sFTP server within Oracle Integration. To use File Server, you must enable it for your Oracle Integration instance. See Enable File Server.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and Oracle ERP Cloud instances in order to connect to these external systems using Oracle Integration, and successfully import financial journal entries from an FTP server to Oracle ERP Cloud.

Access Your FTP Server

Obtain an sFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an output directory (`ftp-erp-transfer`) on the server where the output log files from Oracle ERP Cloud must be uploaded . Note down the path of this directory.

Configure Oracle ERP Cloud

Complete the following configuration tasks on your Oracle ERP Cloud instance to connect to it from Oracle Integration, and import financial journal entries.

Perform the general configuration tasks for creating a connection. See Prerequisites for Creating a Connection.

Financial journal entries have to be in a particular format so that Oracle Enterprise Scheduler (ESS) Jobs can be run and the financial data can be finally stored in a log file. To enable this, you have to download the journal import template from Oracle ERP Cloud and prepare a journal import file based on it.

1. Download the journal import template (`JournalimportTemplate.xlsm`).
   a. Navigate to Oracle Fusion Cloud Applications Suite in Oracle Help Center.
   b. Under **Enterprise Resource Planning (ERP)**, click **Financials**.

c. In the left navigation pane of the Oracle Financials Get Started page, click **All Books**.

d. Under **Implementation**, open the *File-Based Data Import (FBDI) for Financials* guide.

e. Under the **General Ledger** section of the guide, click **Journal Import**.

f. In the **File Links** section, download the journal import template XLSM file (`JournalimportTemplate.xlsm`) and save it to your local machine.

2. Prepare the journal import data file.

a. Delete the sample data from the `JournalimportTemplate.xlsm` file. Keep only one record for your reference.

b. Populate the file with a few financial journal entry records.

c. Then, click and open the Instructions and CSV Generation sheet.

d. Click **Generate CSV File**.

e. Save the generated CSV file. It will be saved as a ZIP file on your local machine.

Upload the data file to the input folder in your FTP server.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select **Yes** from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **FTP Server Access Policy**.
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**  .

Configure Oracle ERP Cloud

Complete the following configuration tasks on your Oracle ERP Cloud instance to connect to it from Oracle Integration, and import financial journal entries.

Perform the general configuration tasks for creating a connection. See Prerequisites for Creating a Connection.

Financial journal entries have to be in a particular format so that Oracle Enterprise Scheduler (ESS) Jobs can be run and the financial data can be finally stored in a log file. To enable this, you have to download the journal import template from Oracle ERP Cloud and prepare a journal import file based on it.

1. Download the journal import template (`JournalimportTemplate.xlsm`).
   a. Navigate to Oracle Fusion Cloud Applications Suite in Oracle Help Center.
   b. Under **Enterprise Resource Planning (ERP)**, click **Financials**.
   c. In the left navigation pane of the Oracle Financials Get Started page, click **All Books**.
   d. Under **Implementation**, open the *File-Based Data Import (FBDI) for Financials* guide.
   e. Under the **General Ledger** section of the guide, click **Journal Import**.
   f. In the **File Links** section, download the journal import template XLSM file (`JournalimportTemplate.xlsm`) and save it to your local machine.
2. Prepare the journal import data file.
   a. Delete the sample data from the `JournalimportTemplate.xlsm` file. Keep only one record for your reference.
   b. Populate the file with a few financial journal entry records.
   c. Then, click and open the Instructions and CSV Generation sheet.
   d. Click **Generate CSV File**.
   e. Save the generated CSV file. It will be saved as a ZIP file on your local machine.

      Upload the data file to the input folder in your FTP server.

Configure the Lookup Table

Edit the **ORCL-BRT-ERP_IMPORTJOURNAL** lookup table and enter appropriate values.

1. In the Lookups section, click the lookup name.
2. Enter appropriate values in the **Property_value** column for the following properties.

Property_name	Property_value
outputDirectoryOnFTP	Enter the path of the output directory that you created on the FTP server. See Access Your FTP Server.
emailTo	Enter the email address where notifications errors have to be sent.
uploadFileName	Enter the name of the log file with the financial journal entries.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Update Integration Properties

The **Oracle ERP FTP Import Journal Callback** integration flow contains the **JobName** integration property that has to updated with appropriate value.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the **JobName** integration property with appropriate value. The **JobName** integration property holds the name of the Oracle Enterprise Scheduler (ESS) job that you run in Oracle ERP Cloud.
3. Click **Submit**.

   A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. the Integrations section of the project workspace, click **Actions** ... on the **Oracle FTP ERP Import Journal** integration flow, then select **Run**.

      You've now successfully submitted the integration for execution. The **Oracle FTP ERP Import Journal** integration flow fetches the financial journal entries from the FTP server in the journal import file and imports them to Oracle ERP Cloud. In Oracle ERP Cloud, after the Oracle Enterprise Scheduler (ESS) job runs successfully, the imported financial journal entries are recorded in a log file.

   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

   c. Now, the **Oracle ERP FTP Import Journal Callback** integration flow gets triggered and it uploads the log file from Oracle ERP Cloud into the specified output folder in the FTP server.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Log in to the FTP server and check if the log file with the imported financial journal entries have been uploaded to the output folder.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using File Server in Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Manage Objects in OCI Object Storage

Use this recipe to manage objects in Oracle Cloud Infrastructure Object Storage with a native action.

Topics:

About This Recipe

This recipe can be used to upload, download, delete or list objects in OCI Object Storage.

To use the recipe, you must install the recipe and configure the resources within the recipe. The recipe uses a native action **OCI Object Storage** to connect to OCI Object Storage service. You don't need to configure an explicit REST adapter connection. The following operations can be used to manage objects in OCI Object Storage:

- **Upload** a file into OCI Object Storage
- **Download** a file from OCI Object Storage
- **List** files in OCI Object Storage
- **Delete** a file in OCI Object Storage

System and Access Requirements

- Oracle Integration 3
- OCI Object Storage
- Access to an OCI Object Storage bucket

Before You Install the Recipe

Create a dynamic group and policies to use the OCI Object Storage native action.

See **Prerequisites** in Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources.

Configure the Lookup Table

Edit the lookup table and enter appropriate values for the lookup keys.

1. In the Lookups section, click the lookup name.
2. Enter appropriate values for the look up keys.

Key	Value
BucketName	Specify the name of the OCI Object Storage bucket where you want to manage objects.
EmailFrom	Enter the email address from which error notifications are to be sent.
EmailTo	Enter the email address to which error notifications are to be sent.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Test the recipe in Oracle Integration.
   a. In the Integrations section of the project workspace, click **Actions** on the **Oracle Object Storage Handler** integration flow, then select **Run**.
   b. On the Configure and run page, in the Request section, select **File** and then upload the object to add to OCI Object Storage (for example, a jpg file).
   c. Click **Run**
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Check if the object (jpg file) has been added to the specified bucket in OCI Object Storage.
   a. Go to Oracle Cloud Infrastructure Console.
   b. In the navigation pane, select **Storage** and then **Buckets**.
   c. Select the bucket and check if the file that you uploaded is displayed under the Objects section.

Related Documentation

- *OCI Object Storage documentation*

Manage ServiceNow Incidents

Use this recipe to create, update, or delete incidents in ServiceNow.

Topics:

About This Recipe

This recipe can be used to create an incident, update an incident, or delete an incident in ServiceNow.

To use the recipe, you must install the recipe and configure the connections within it. The recipe contains three integrations: **Oracle REST ServiceNow Create Incident**, **Oracle REST ServiceNow Update Incident**, and **Oracle REST ServiceNow Delete Incident**.

- When the **Oracle REST ServiceNow Create Incident** integration flow is triggered by a REST request with `incidentdescription` as a request parameter, the recipe creates an incident in ServiceNow. The response containing the incident number and the incident Id is returned to you.
- When the **Oracle REST ServiceNow Update Incident** integration flow is triggered by a REST request with `incidentId`, `incidentState`, and `incidentUrgency` as request parameters, the recipe updates the corresponding incident details in ServiceNow.
- When the **Oracle REST ServiceNow Delete Incident** integration flow is triggered by a REST request with `incidentId` as a request parameter, the recipe deletes an incident in ServiceNow.

Integration	Description	REST Commands and Example Payloads

Oracle REST ServiceNow Create Incident	Creates an incident in the ServiceNow instance.	**REST API Command:**
		
		`POST: https://<host:port>/ic/api/integration/v2/flows/rest/project/ORCL-R-REST_SNOW_MANAGE_INCIDENT/ORCL-R-REST_SERVIC_UPDATE_INCIDE/1.0/create`
		
		Example Request Payload:
		
		`{ "incidentdescription" : "Employee phone number missing" }`
		
		Example Response Payload:
		
		`{ "serviceNowNr": "INC0010031", "serviceNowSysId": "fe6a74e62f503010f129811df699b6e7" }`

Integration	Description	REST Commands and Example Payloads

Oracle REST ServiceNow Update Incident	Updates an incident in the ServiceNow instance.	**REST API Command:**
		`POST: https://<host:port>/ic/api/integration/v2/flows/rest/project/ORCL-R-REST_SNOW_MANAGE_INCIDENT/ORCL-R-REST_SERVIC_UPDATE_INCIDE/1.0/update`
		Example Request Payload:
		`{"incidentId":"bec4ce452f603010f129811df699b6cc","incidentState":"2","incidentUrgency":"2"}`
		Note the valid values of request parameters incidentState and incidentUrgency:
		`incidentState`
		• 1 = New
		• 2 = In Progress
		• 3 = On Hold
		• 6 = Resolved
		• 7 = Closed
		• 8 = Canceled
		`incidentUrgency`
		• 1 = High
		• 2 = Medium
		• 3 = Low
		Example Response Payload:
		`{ "incidentStatus": "bec4ce452f603010f129811df699b6cc" }`

Integration	Description	REST Commands and Example Payloads

Oracle REST ServiceNow Delete Incident	Deletes an incident in the ServiceNow instance.	**REST API Command:**
		`POST: https://<host:port>/ic/api/integration/v2/flows/rest/project/ORCL-R-REST_SNOW_MANAGE_INCIDENT/ORCL-R-REST_SERVIC_UPDATE_INCIDE/1.0/delete`
		Example Request Payload:
		`{"incidentId":"fe6a74e62f503010f129811df699b6e7"}`
		Example Response Payload:
		`{ "status": "1" }`

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- ServiceNow
- An account on ServiceNow with the Administrator role

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle ServiceNow Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the ServiceNow instance URL in the **ServiceNow Instance Name** field. For example: `https://host_name.service-now.com`
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	By default, **Basic Authentication** is selected.
Username	Enter the user name of your ServiceNow instance.
Password	Enter the password of your ServiceNow instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe from an external application.
   a. In the Integrations section of the project workspace, click **Actions** on the integration flow, then select **Run**.
   b. On the Configure and run page, click **Endpoint metadata**.
   c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
   d. From the external application, send the required REST command to this endpoint URL along with the request parameters. See **REST Commands and Example Payloads** in About This Recipe.
3. Test the recipe in Oracle Integration.
   a. In the Integrations section of the project workspace, click **Actions** on the integration flow, then select **Run**.
   b. On the Configure and run page, in the Request section, under **URI parameters**, enter the request parameters.
   c. Click **Run**

      The Activity stream pane appears displaying the status of the integration instance's execution.

d. In the Response section of the Configure and run page, you'll find a success response `200 OK`.

4. View the response payload in the Activity stream to verify if the action is successful. For example, if you had sent a `POST` request with the required request parameters to create an incident in ServiceNow, check if the incident number and incident Id is returned in the response.

Related Documentation

- *Using the ServiceNow Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Migrate Integration Artifacts Between Instances

Use this recipe to migrate integration artifacts (such as, integration flows, connections, lookups, libraries, agent groups, and so on) between two Oracle Integration instances.

Topics:

About This Recipe

This recipe migrates integration artifacts (integration flows and associated resources) between two Oracle Integration instances. For example, you can use this recipe to migrate integration artifacts from the development environment to the testing environment.

The recipe performs the following tasks:

1. Checks the status of the source and target instances. If they are not reachable, the recipe terminates the migration process.
2. Executes a loop on the integration artifacts to migrate them if the instances are reachable.
3. Exports integration artifacts from the source instance to the target instance.
4. Sends the migration report to the configured email address.

System and Access Requirements

- Oracle Integration Source Instance
- Oracle Integration Target Instance

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Connection (Trigger)

1. In the Connections section, click the connection name.
2. In the Security section, select the Security Policy as either **OAuth2.0** or **Basic Authentication**.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
5. To return to the project workspace, click **Go back**  .

Configure the Oracle REST Source Connection (Invoke)

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave **REST API Base URL** selected.
Connection URL	Enter the Oracle Integration design time URL of the source instance. To obtain the design time URL, paste the service console URL in a new browser window. When the log in page of the Oracle Integration instance displays, note that the URL changes to a design time URL. All design time URLs start with the word `design` as shown in the following example: `https://design.integration.region.ocp.oraclecloud.com/?integrationInstance=NameOfServiceInstance` Enter only the following part of the URL: `https://design.integration.region.ocp.oraclecloud.com`

3. In the Security section, select the Security Policy as either **OAuth Client Credentials** or **OAuth Authorization Code Credentials**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Oracle REST Target Connection (Invoke)

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave **REST API Base URL** selected.

Field	Information to Enter
Connection URL	Enter the Oracle Integration design time URL of the source instance. To obtain the design time URL, paste the service console URL in a new browser window. When the log in page of the Oracle Integration instance displays, note that the URL changes to a design time URL. All design time URLs start with the word `design` as shown in the following example: `https://design.integration.region.ocp.oraclecloud.com/?integrationInstance=NameOfServiceInstance` Enter only the following part of the URL: `https://design.integration.region.ocp.oraclecloud.com`

3. In the Security section, select the Security Policy as either **OAuth Client Credentials** or **OAuth Authorization Code Credentials**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Lookup Table

Edit the **Oracle_Integration_Artifacts_Migration_Utility** lookup table.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

Key	Description
IntegrationStatusAPI	The API URL to check the status of the Oracle Integration instance.
EmailTo	The email address to which the email notifications are sent. Enter an email address of your choice.
EmailFrom	The email address from which the email notifications are sent. Enter an email address of your choice.
SourceIntegrationInstance	The source Oracle Integration service instance name.
TargetIntegrationInstance	The target Oracle Integration service instance name.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Test the recipe in Oracle Integration.

a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

b. On the Configure and run page, in the Request section, click **Body** and enter the payload with the following details:

Element	Value to Migrate Integrations
artifactType	Enter `integrations`.
code	Enter the integration identifier.
version	Enter the integration version. For example: `01.00.0000`

For example:

```
{
  "artifact": [
    {
      "artifactType": "integrations",
      "code": "GETCURRENT",
      "version": "01.00.0000"
    }
}
```

c. Click **Run**

The integration flow now automatically migrates the integration artifacts form the source instance to the target instance. The Activity Stream pane appears displaying the status of the integration instance's execution.

d. In the Response section of the Configure and run page, under the **Body** tab, you'll find a HTTP status code as **202 accepted**.

3. Log in to your target instance and check for the migrated integration artifacts.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Post Slack Notifications for Completed SurveyMonkey Surveys

Use this recipe to post a notification message in Slack when a survey created with SurveyMonkey is completed by a respondent.

Topics:

About This Recipe

This recipe posts notification messages to a specified Slack channel for completed SurveyMonkey surveys, according to a specified schedule in Oracle Integration. Using the recipe you can track survey responses by getting notification messages in Slack with details about completed survey responses.

The recipe uses the standard SurveyMonkey Adapter and the standard Slack Adapter. To use the recipe, you must install the recipe and configure connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered the integration queries for completed surveys in SurveyMonkey, and posts notification messages for the completed surveys to the specified Slack channel.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39462) or higher
- SurveyMonkey
- An account in SurveyMonkey with the Administrator role
- Slack
- An account in Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SurveyMonkey instance and Slack instance in order to successfully connect to these external systems using Oracle Integration and post notification messages in Slack for completed survey responses.

Configure SurveyMonkey

You must perform the following prerequisite tasks in your SurveyMonkey instance.

1. Create and configure a SurveyMonkey app.
    a. Log in to the SurveyMonkey Developer console at https://developer.surveymonkey.com/.
    b. Click **My Apps** on the title bar of the SurveyMonkey Developer home page.
    c. On the My Apps page, click **Add a New App**.
    d. In the resulting APP CREATION dialog, enter the app's name in the **App Nickname** field, and select **Private App** as the **App Type** option.
    e. Click **Create App**.

       SurveyMonkey creates an app and displays the client ID and secret details on the resuling page.
    f. Under the Settings section, enter the following in the **OAuth Redirect URL** field.

       ```
       https://{OIC_HOST}:{OIC_SSL_PORT}/icsapis/agent/oauth/callback
       ```

       Replace `OIC_HOST` and `OIC_SSL_PORT` with specific values to your Oracle Integration instance.
    g. Get the SurveyMonkey API credentials.
        i. Click **SETTINGS** to open the app's settings.
        ii. Scroll down to the Credentials section, and note the values in the **Client ID** and **Secret** field.

Click the eye icon in the **Secret** field to view the value. You'll need the SurveyMonkey API credentials, that is the client ID and secret values while configuring the SurveyMonkey connection from Oracle Integration.

h. Select the scopes.

i. In the Scopes section, select the following scopes:

- Create/Modify Surveys
- Create/Modify Collectors
- Create/Modify Responses
- View Surveys
- View Collectors
- View Responses
- View Response Details

ii. Click **Update Scopes**.
You get a confirmation message that the scopes are successfully updated.

2. Create a survey in SurveyMonkey for which you want to receive responses.

a. Log in to your SurveyMonkey account at https://www.surveymonkey.com/user/sign-in/.

b. On the SurveyMonkey Dashboard, click **CREATE SURVEY** on the title bar.

c. Create a survey using one of the following options in the Create a new survey pane on the left:

- Start from scratch
- Copy a past survey
- Buy Targeted Response
- Import Questions
- Start from template
- Build it for me

Find information related to creating and managing surveys under the Design and Manage topic in the SurveyMonkey Help Center (https://help.surveymonkey.com/). Note down the name of the survey that you have created.

d. Click **Next** to preview your survey.

e. Click **Next** to view the COLLECT RESPONSES section.

f. Click **Send surveys your way**.

The COLLECT RESPONSES section displays the different ways in which you can collect the responses.

g. Click **Share a survey link**.

The COLLECT RESPONSES section displays the details of the web link collector. Note down the name of the collector. For example, `Web Link 1`.

Configure Slack

Complete the following tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

1. Create a Slack app and configure it to receive notifications for successful Stripe charges.

a. Log in to the Slack app platform at https://api.slack.com.

b. From the workspace navigation menu, click your workspace.

c. Select **Administration** and then select **Manage apps**.

d. In the Slack app directory page, click **Build**.

e. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.

f. In the Name app & choose workspace dialog, enter the following details.

Field	Information to Enter
App Name	Enter the name of your app. For example, `Oracle Integration App`.
Pick a workspace to develop your app in	Select your workspace from the drop-down list.

g. Click **Create App**.

The app gets created and its Basic Information page appears.

h. Get the Slack App API credentials. In the app's Basic Information page, scroll down to the App Credentials section. Note the client ID in the **Client ID** field, and the client secret in the **Client Secret** field.

These API credentials, that is the client ID and secret, were automatically generated when you created the app. You'll use these later while configuring connections to your Slack instance from Oracle Integration.

i. Set permission and scope for your Slack App. In the left menu, under **Features**, select **OAuth & Permissions**.

j. In the OAuth & Permissions page, go to the Redirect URLs section.

k. Click **Add New Redirect URL** and enter the URL of your Oracle Integration instance. For example, `https://your_instance_URL:443/icsapis/agent/oauth/callback`, where `your_instance_URL` is the Host URL of your Oracle Integration instance.

l. Click **Add**, and then click **Save URLs**.

m. Scroll to the Scopes section, and under User Token Scopes click **Add an OAuth Scope**.

n. Enter the following scopes:

- channels:read
- channels:write
- chat:write
- groups:read
- groups:write
- usergroups:write

Note the scopes. You'll use these later while configuring connections to your Slack instance from Oracle Integration.

o. Press Enter.

2. You have to set up a workspace in Slack, create a channel in the workspace, and then add people who should receive the survey completed notification messages into the channel.

a. Create a workspace in Slack.

i. Open the Slack get started page using the following URL:

```
https://slack.com/get-started#/createnew
```

ii. Enter your email and click **Continue**.

iii. Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.

b. Create a channel to post notifications.

i. On your Slack instance, select your workspace.

ii. Click the workspace name on the left navigation pane, and from the menu that displays, select **Create a channel**.

iii. In the Create a channel dialog, enter your channel's name in the **Name** field. For example, `#survey-responses`.

iv. Optionally, enter a suitable description in the **Description** field to let others know what the channel is about.

v. Click **Create**.

c. Add people to your channel.

i. On your Slack workspace, select the channel from the left navigation pane to open it.

ii. Click **Add people** on the top right of the channel.

iii. In the resulting Add people dialog, enter names, emails, or user groups of people you want to add in the field provided.

iv. Click **Add**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle SurveyMonkey Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID of the OAuth application created in SurveyMonkey. See Configure SurveyMonkey.
Client Secret	Enter the client secret of the OAuth application created in SurveyMonkey.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

You'll now be redirected to the SurveyMonkey login page.

6. Enter your SurveyMonkey account credentials and click **Log In**.

   You are now displayed a page that prompts you to authorize the connection to the SurveyMonkey application created previously.

7. On the SurveyMonkey web page, click **Authorize**.

   You're informed that access to SurveyMonkey is allowed. You can now switch back to the SurveyMonkey Connection window of Oracle Integration to test your connection.

8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

9. To return to the project workspace, click **Go back** .

Configure the Oracle Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Configure Slack.
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter the permission scopes you configured while creating and configuring the Slack app.

3. Click **Provide Consent**.

   A new browser window opens to approve access to Slack.

4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Update the integration property values.
   a. In the Integrations section of the project workspace, Hover over the **Oracle SurveyMonkey Slack Post Completed Responses** integration flow, then click **Actions** ... and select **Update Property Values**.
   b. In the Update Property Values dialog, note the properties that have been configured - **ChannelName**, **SurveyName**, and **NotificationMail**.

c. Click **ChannelName** and in the **New Value** field, enter the Slack channel name where you want the notification messages to be posted for completed survey responses.

   Optionally, you can also enter the value of the Slack channel ID. To obtain the Slack channel ID, see Configure Slack.

d. Click **SurveyName** and in the **New Value** field, enter the name of the survey that you had created in Slack. See Configure Slack.

e. Click **NotificationMail** and in the **New Value** field, enter the email ID where you want to notify in case a runtime error/exception occurs while running the integration.

f. Click **Submit**.

   A message confirms that the integration properties have been updated successfully.

3. Complete the survey.

   a. Log in to your SurveyMonkey account.

   b. Click **My Surveys** on the title bar.

   c. Select the survey that you created.

   d. Navigate to **COLLECT RESPONSES**, and use one of the collectors to participate in the survey.

      For the purpose of testing, we'll choose the **Web link collector** option from the **ADD NEW COLLECTOR** drop-down list. Click the collector from the Survey Collector table, and in the resulting page that opens click **COPY** besides the web link URL. Copy and paste the link to a new browser window.

   e. Access and complete the survey.

4. Run the recipe.

   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

5. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.

6. Check if a notification is posted in the Slack channel.

   a. In Slack, navigate to the channel created for posting completed survey response notifications.

   b. Check if the notification message for the survey that you completed in step 3 is posted in the channel.

Related Documentation

- *Using the SurveyMonkey Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

Post Slack Notifications for New Marketo Leads

Use this recipe to post notification messages in Slack for new Marketo leads.

Topics:

About This Recipe

This recipe posts notification message to a specified Slack channel with details about new leads that are created in Marketo. It uses the standard Marketo Adapter and the Slack Adapter.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow manually or specify an execution schedule for it. When triggered, the integration flow queries for list of leads in Marketo. For the first time, it fetches all leads present in the Marketo instance, creates a message with the lead details (name and company), and posts the message to the specified Slack channel. For subsequent executions, it filters out the old leads and fetches only the new leads created since the last execution of the integration flow, and then posts notification messages to the specified Slack channel with details about the new leads.

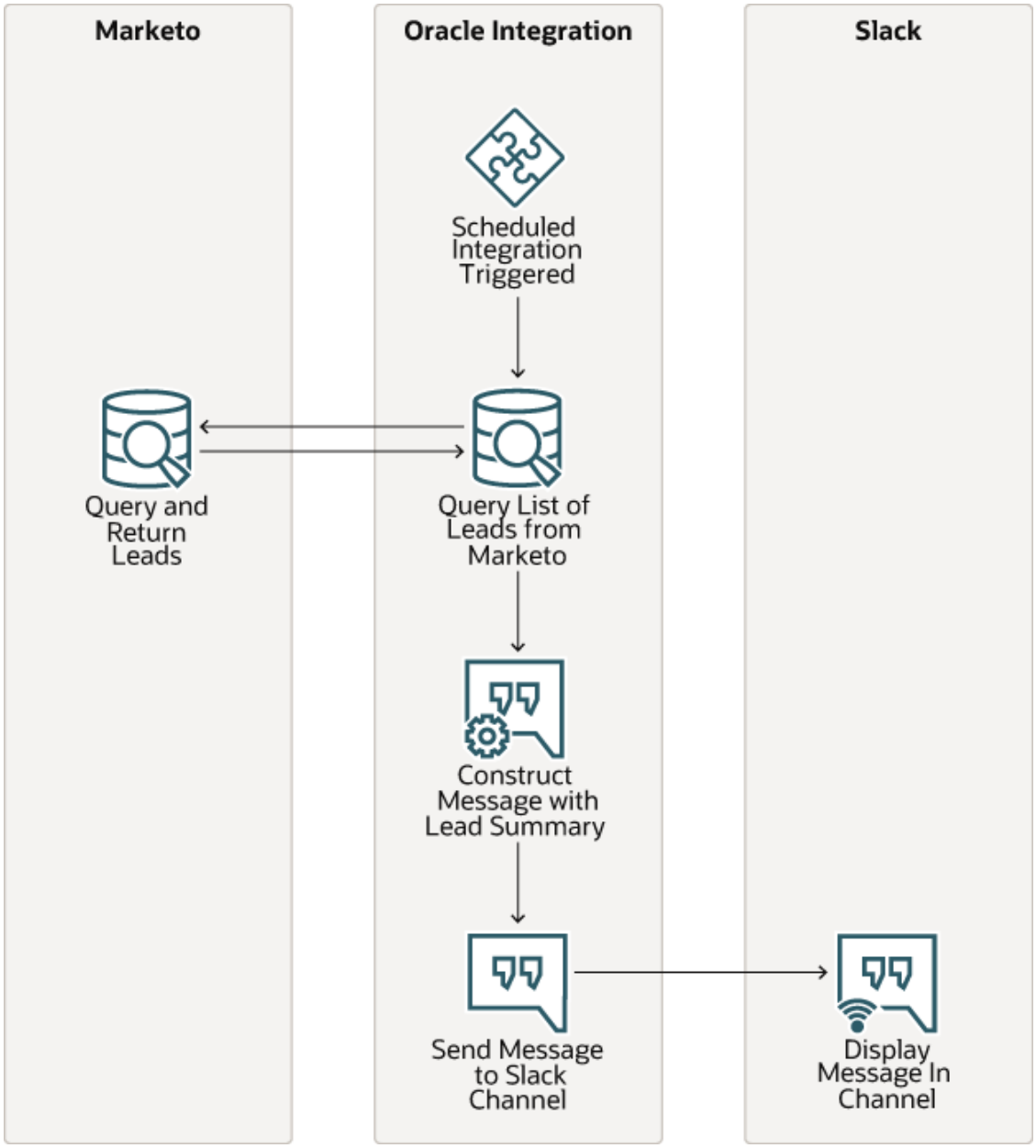
System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Marketo
- Slack
- An account in Marketo with the Administrator role
- An account in Slack with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.

When the integration flow of the recipe is triggered by an execution schedule or manual submission, it queries the Marketo instance for a list of leads. It filters out the old leads, that is, the leads that were already retrieved in the last integration flow; retrieves the new leads from the list, and then creates a message with details about the new leads. This message with details about the new leads is then sent to Slack and posted to the specified Slack channel.



Before You Install the Recipe

You must perform the following configuration tasks on your Marketo and Slack instances in order to connect to these external systems using Oracle Integration and successfully post notification messages in Slack for new Marketo leads.

Configure Marketo

Complete the following tasks in your Marketo instance to successfully connect to it from Oracle Integration.

1. Create an API only user role.
2. Create a customer service.
3. Obtain the Client ID and Client Secret.

4. Obtain the Munchkin ID.

For detail information on the above steps, see Prerequisites for Creating a Connection in *Using the Marketo Adapter with Oracle Integration 3*.

Configure Slack

Complete the following tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

Create and Configure a Slack App

1. Create a Slack App.
   a. Log in to the Slack app platform at https://api.slack.com.
   b. In the Slack API title bar, click **Your Apps**.
   c. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
   d. In the Name app & choose workspace dialog, enter the following details.

Field	Information to Enter
App Name	Enter the name of your app. For example, `Oracle Integration App`.
Pick a workspace to develop your app in	Select your workspace from the drop-down list.

   e. Click **Create App**.
   The app gets created and its Basic Information page appears.
2. Get the Slack App API credentials.
   a. In the app's Basic Information page, scroll to the App Credentials section.
   b. Note the client ID in the **Client ID** field, and the client secret in the **Client Secret** field. These API credentials, that is the client ID and secret, were automatically generated when you created the app. You'll use these later while configuring connections to your Slack instance from Oracle Integration.
3. Set permission and scope for your Slack App.
   a. In the left menu, under **Features**, select **OAuth & Permissions**.
   b. In the OAuth & Permissions page, go to the Redirect URLs section.
   c. Click **Add New Redirect URL** and enter the URL of your Oracle Integration instance. For example, `https://your_instance_URL:443/icsapis/agent/oauth/callback`, where `your_instance_URL` is the Host URL of your Oracle Integration instance.
   d. Click **Add**, and then click **Save URLs**.
   e. Scroll to the Scopes section, and under User Token Scopes click **Add an OAuth Scope**.
   f. Enter `chat:write`.
   g. Press Enter.

Set Up Your Slack WorkSpace and Channel

You have to set up a workspace in Slack, create a channel in the workspace, and then add people into the channel who should receive notification messages for events in the Amazon S3 bucket.

1. Create a workspace in Slack.
    a. Open the Slack get started page using the following URL:

       `https://slack.com/get-started#/createnew`

    b. Enter your email and click **Continue**.
    c. Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.
2. Create a channel to post notifications.
    a. On your Slack instance, select your workspace.
    b. Click the workspace name on the left navigation pane, and from the menu that displays, select **Create a channel**.
    c. In the Create a channel dialog, enter your channel's name in the **Name** field. For example, `#marketo-leads`.
    d. Optionally, enter a suitable description in the **Description** field to let others know what the channel is about.
    e. Click **Create**.
3. Add people to your channel.
    a. On your Slack workspace, select the channel from the left navigation pane to open it.
    b. Click **Add people** on the top right of the channel.
    c. In the resulting Add people dialog, enter names, emails, or user groups of people you want to add in the field provided.
    d. Click **Add**.

Get the Slack Channel ID

You must get the channel ID of the channel that you created on your Slack workspace. You'll use the channel ID for configuring the integration properties in Oracle Integration, so that notifications are sent to the right Slack channel.

1. On your Slack workspace, navigate to the channel under **Channels** in the left navigation pane.
2. Right-click the channel name.
3. In the menu that displays, click **Copy link**.
4. Copy the link to a text editor, such as Notepad.
5. Note the nine characters after the last forward slash (/) in the link. This is the channel ID of the Slack channel. Note that the channel ID can be a combination of letters and numbers. For example, CB64YSB2D.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle Marketo Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:
   - In the **Munchkin ID** field, enter the munchkin ID that you obtained earlier. See Configure Marketo.
3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID obtained earlier. See Configure Marketo.
Client Secret	Enter the client secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Oracle Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Configure Slack.
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter `chat:write:bot`.

3. Click **Provide Consent**.

   A new browser window opens to approve access to Slack.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

7. To return to the project workspace, click **Go back**  .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions** ⋮ on the **Oracle Marketo Slack Schd New Lead Notification** integration flow, then select **Update property values**.
2. In the Update property values dialog, update the integration properties with appropriate values.

   The recipe's integration flow contains the following properties. Click an integration property and update its value in the **New Value** field.

   a. **SlackChannelId**: This integration property holds the channel ID to which notifications are sent for new Marketo leads. Enter the channel ID of your Slack channel. See Configure Slack.

   b. **NumberOfLeadIds**: This integration property holds the total number of Marketo leads for which you want to send Slack notifications. Enter the number of lead information that you want to retrieve from Marketo. The default value is 10.
3. Click **Submit**.

   A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

> **Note:**
>
> Before activating and running the recipe, ensure that there are new leads in your Marketo instance.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.

   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

   The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

   - **LastExecutionDate**: This parameter stores the date of the most-recent successful run of the integration flow. For the initial run, it contains a default value. The parameter's value is automatically updated after each successful run, and only the Marketo leads created after the date stored as the parameter's **Current value** are processed by the integration in each run. If you want to change the date for a specific scenario, enter the date of your choice in the **New value** field in the format, yyyy-mm-dd.
   - **NewestLeadId**: This parameter stores the Id of a Marketo lead. Only the Marketo leads created after the LeadId stored in this parameter are processed by the integration in each run. Enter a new LeadId of your choice from Marketo in the **New value** field.

b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

> **Note:**
>
> You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Check if a notification message is posted in Slack.
   a. In your Slack instance, navigate to the channel that you created for posting notifications for new Marketo leads.
   b. Check if a notification message with the list of new leads from Marketo is posted to the channel.

Related Documentation

- *Using the Marketo Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

Post Slack Notifications for Oracle CPQ Quote Approvals

Use this recipe to post notifications in Slack for Oracle Configure, Price, and Quote Cloud (Oracle CPQ) quote approvals.

Topics:

About This Recipe

This recipe posts a notification message to a specified Slack channel when an Oracle CPQ quote is sent for approval. The Slack notification message also contains a link to access the transaction approval page.

To use the recipe, you must install the recipe and configure connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe. When a quote is sent for approval in Oracle CPQ, the integration flow gets triggered, and it subsequently posts a notification message in the specified Slack channel with a link to the transaction approval page for the quote.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle CPQ
- Slack

- An account on Oracle CPQ with the Administrator role
- An account on Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle CPQ and Slack instances in order to successfully connect to these external systems using Oracle Integration and post Slack notifications for Oracle CPQ quote approvals.

Configure Oracle CPQ

To access Oracle CPQ using Oracle Integration and ensure that notifications are posted in Slack for quote approvals, you must perform certain configuration tasks in your Oracle CPQ instance.

Log in to your Oracle CPQ instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration with the **Web Services Only** permission.
   a. On the Oracle CPQ home page, click **Admin Drawer** on the title bar.
   b. In the Admin navigation pane, expand **Users** and click **Internal Users**.

      The User Administration List page appears.
   c. Add the user from the User Administration List page.

      For details, see Setting Up Users.
2. Obtain the REST Catalog URL. See Prerequisites for Creating a Connection.
3. Create an integration in Oracle CPQ to establish a connection from Oracle CPQ to Oracle Integration.
   a. In the Admin navigation pane, expand **Integration Platform** and click **Integration Center**.
   b. On the Integration Center page, click **Create Integration**.
   c. From the **Type** drop-down list, select **Integration Cloud Service**.
   d. In the resulting page enter the following details, and click Test to verify if your connection works.

Fields	Information to Enter
Name	Enter a suitable name.
Variable Name	Enter a variable name.
Discovery URL	Enter the Oracle Integration instance URL.
Username	Enter the username to access your Oracle Integration instance.
Password	Enter the password to access your Oracle Integration instance.

   e. Select **Enable Integration**, and then click **Save**.
4. Link the integration that you created to a process.
   a. In the Admin navigation pane, expand **Commerce and Documents** and click **Process Definition**.
   b. On the Processes page, select the process to which you want to link the integration. Select **Integrations** from the **Navigation** drop-down list, and click **List**.

c. On the resulting Integrations page, click **Add**.

d. In the Select Integration Types page, select **Integration Cloud Service** and click **Next**.

e. On the Edit Integration page, enter the following details and click **Add**.

Fields	Information to Enter
Name	Enter a suitable name.
Variable Name	Enter a variable name.
Timeout	Enter the timeout in milliseconds.
Description	If required, enter a suitable description.
Action	Select the **Export** option.
Services	Select the Service.

5. Specify the action that triggers the integration.

a. On the Processes page, select the process linked to the integration. Select **Documents** from the **Navigation** drop-down list, and click **List**.

b. In the **Transaction** row of the Document List page, select **Actions** from the **Navigation** drop-down list, and click **List**.

c. On the Actions List page, select **Submit** and then **Request Approval**.

d. On the Admin Action page, click the **Integration** tab.

e. From the **Integration List** box, select the integration created in Step 3 and click the **>** arrow to add it to the **Selected Integration** box.

f. Click **Update**.

6. Specify the approval rule and the user that will approve the quote.

a. On the Processes page, select the process linked to the integration. Select **Integrations** from the **Navigation** drop-down list, and click **List**.

b. In the **Transaction** row of the Document List page, select **Actions** from the **Navigation** drop-down list, and click **List**.

c. On the Actions List page, select **Submit**.

d. Click the **General** tab, and navigate to **Approval Sequence** and **Edit User Approval**.

e. Click **+**, enter the **Reason Name** and **Variable Name**, and then click **Save**.

f. Select **Simple Condition**, specify the **Attribute Name**, **Operator**, and **Attribute Value**, and then click **Save**.

g. Click **Reason Flow**, click **Approvals**, and then select the approver who would approve the quote from the drop-down list.

Configure Slack

To access Slack using Oracle Integration, and ensure that Oracle CPQ quote approval notifications are posted in the specified Slack channel, you must perform certain configuration tasks in your Slack instance.

1. Create a Slack workspace.

a. Open the Slack get started page using the following URL:

```
https://slack.com/get-started#/createnew
```

b. Enter your email and click **Continue**.

c. Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.

2. Create a channel to post notifications.

a. On your Slack instance, select your workspace.

b. Click the workspace name on the left navigation pane, and from the menu that displays, select **Create a channel**.

c. In the Create a channel dialog, enter your channel's name in the **Name** field. For example, `#cpq-quote-approval`.

d. Optionally, enter a suitable description for the channel in the **Description** field.

e. Click **Create**.

3. Add people to your channel.

a. On your Slack workspace, select the channel from the left navigation pane to open it.

b. Click **Add people** on the top right of the channel.

c. In the resulting Add people dialog, enter names, emails, or user groups of people you want to add in the field provided.

d. Click **Add**.

4. Create a Slack App.

a. Log in to the Slack app platform at `https://api.slack.com`.

b. In the Slack API title bar, click **Your Apps**.

c. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.

d. In the Name app & choose workspace dialog, enter the following details:

i. In the **App Name** field, enter the name of your app.
For example, `Oracle Integration App`.

ii. In the **Pick a workspace to develop your app** in field, select your workspace from the drop-down list.

e. Click **Create App**.

The app gets created and its Basic Information page appears.

5. Get the Slack App API credentials.

a. In the app's Basic Information page, scroll down to the App Credentials section.

b. Note the client ID in the **Client ID** field, and the client secret in the **Client Secret** field.

You'll use these later while configuring connections to your Slack instance from Oracle Integration.

6. Set permissions for your Slack App.

a. In the left menu, under **Features**, select **OAuth & Permissions**.

b. In the OAuth & Permissions page, go to the Redirect URLs section.

c. Click **Add New Redirect URL** and enter the URL of your Oracle Integration instance.

For example, `https://your_instance_URL:443/icsapis/agent/oauth/callback` where *your_instance_URL* is the Host URL of your Oracle Integration instance.

d. Click **Add**, and then click **Save URLs**.

e. Scroll down to the Scopes section, and under User Token Scopes click **Add an OAuth Scope**.

f. Enter the following scopes:

- channels:read
- channels:write
- groups:read
- groups:write
- usergroups:write

Note the scopes. You'll use these later while configuring connections to your Slack instance from Oracle Integration.

g. Press Enter.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle CPQ Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select **REST Catalog URL**. See Configure Oracle CPQ.
Connection URL	Enter the instance URL of Oracle CPQ.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
User Name	Enter the user name of the account created for Oracle Integration in Oracle CPQ. See Configure Oracle CPQ.
Password	Enter the password of the account created for Oracle Integration in Oracle CPQ.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**  .

Configure the Oracle Slack Connection

1. In the Connections section, click the connection name.

2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID you obtained earlier while creating and configuring a Slack app. See Configure Slack.
Client Secret	Enter the client secret you obtained earlier while creating and configuring a Slack app.
Scope	Enter the permission scopes you configured while creating and configuring a Slack app.

3. Click **Provide Consent**.

   A new browser window opens to approve access.

4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Update Integration Property

You have to update the integration property value so that Oracle CPQ quote approval notification messages are sent to the right Slack channel.

1. In the Integrations section, click **Actions** on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, click **SlackChannelId**.
3. In the **New Value** field, enter the name of the Slack channel where you want the notifications to be posted.
4. Click **Submit**.

   You get a confirmation message that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. Log in to your Oracle CPQ instance.
   b. Create a transaction and specify the quote details.

      Note that you can only submit a quote for approval, if it fulfills the specified approval rule.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.

4. Check if a notification message is posted in the Slack channel.
    a. In your Slack instance, navigate to the channel you created for posting notifications for Oracle CPQ quote approvals.
    b. Check if a notification message with a link to the Oracle CPQ quote approval page is posted to the channel.
5. Check if you can access the Oracle CPQ quote approval page and approve the quote as the approver.
    a. Click the link in the Slack notification message.
    b. Log in to Oracle CPQ as the approver.
    c. On the Approval Status page, enter Approved in the **My Approval** field. A green check mark indicates that the quote has been approved successfully.

       The Approval History section displays details such as the user who approved the quote and the date on which the quote is approved.

Related Documentation

- *Using the Oracle CPQ Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

Send Automatic Replies to Emails with a Specific Subject Using Oracle AI

Use this recipe to send automatic replies to Microsoft Office 365 Outlook emails with a specific subject using Oracle AI.

Topics:

About This Recipe

This recipe retrieves Microsoft Office 365 Outlook emails with specific subject and automatically sends responses to such emails by leveraging the artificial intelligence capabilities of Oracle AI.

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe. The recipe contains three integration flows. The first integration flow **Oracle Get Email** runs at a defined schedule and fetches unread emails from the Outlook server to Oracle Integration. It filters emails with specific subjects from the retrieved emails and publishes them to the Publish event in JSON format. This in turn triggers the second integration flow **Oracle GenAI Quick ReplyToEmail** which subscribes to the Publish event. The second integration flow uses OCI Generative AI capabilities to generate a response to the email, which is then included in the body of the reply email. It also triggers the third integration flow **Oracle MaskingPIIfromEmail** that leverages OCI Language AI capabilities - specifically `BatchDetectLanguageEntities`, to identify key entities (for example: names, dates, locations, account numbers) in the email content, and masks sensitive data in the response.

Subsequently, the AI generated response is sent to the email sender by the **Oracle GenAI Quick ReplyToEmail** integration flow.

System and Access Requirements

- Oracle Integration 3
- OCI Language AI
- OCI Generative AI

Before You Install the Recipe

You must perform the following configuration tasks on your Microsoft, OCI Language AI Service, OCI Generative AI Service instances in order to successfully connect to these systems using Oracle Integration and send automatic responses to emails.

1. Configure Microsoft Office 365 Outlook. See Prerequisites for Creating a Connection.
2. Access and configure OCI Language AI Service.

   See the following topics in OCI documentation:

   - Overview of Language AI Service
   - Batch Detect Language Entities API
   - Personal Identifiable Information
3. Access and configure OCI Generative AI Service. See Overview of Generative AI Service.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Microsoft Graph REST API Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection type	Select **REST API Base URL**.
Connection URL	Enter the connection URL. For example: `https://graph.microsoft.com/v1.0`

3. In the Security section, enter the following details:

Field	Information to Enter
Security policy	Select **OAuth Authorization Code**.
Client Id	Enter the client Id.
Client Secret	Enter the client secret.

Field	Information to Enter
Authorization Code URI	Enter `https://login.microsoftonline.com/<tenantId>/oauth2/v2.0/authorize`
Access Token URI	Enter `https://login.microsoftonline.com/<tenantId>/oauth2/v2.0/token`

4. In the **Scope** field, enter the scope URL.

   For example:

   ```
   https://graph.microsoft.com/Mail.ReadWrite https://graph.microsoft.com/Mail.Send offline_access
   ```

5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

7. To return to the project workspace, click **Go back**.

Configure the Microsoft Outlook Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client Id.
Client Secret	Enter the client secret.
Scope	Enter the scope URL. For example `https://graph.microsoft.com/Mail.ReadWrite https://graph.microsoft.com/Mail.Send offline_access`

3. Click **Provide Consent**. This enables Oracle Integration to interact with the Microsoft Office 365 Outlook account used to create the application at `https://portal.azure.com/`. If everything is correct, you are prompted for the Oracle Integration credentials.
   a. Enter the credentials and click **OK**. These are the same credentials you use to log in to Oracle Integration.
   b. Enter the Microsoft Office 365 Outlook account credentials.

      A page is displayed asking for permission to interact with the account.

   c. Click **Yes**.

      The Access Allowed! page is displayed.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Configure the OCI Language AI REST API Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection type	Select **REST API Base URL**.
Connection URL	Enter the Language AI Service API endpoint to use. For example: `https://language.aiservice.us-ashburn-1.oci.oraclecloud.com/20221001`

3. In the Security section, enter the following details:

Field	Information to Enter
Security policy	Select **OCI Signature Version 1**.
Tenancy OCID	Enter your tenancy OCID. See Finding Your Tenancy OCID (Oracle Cloud Identifier)
User OCID	Enter your user OCID. See Resource Identifiers.
Private Key	Enter the private key. See *How to Generate API Signing Key* in Required Keys and OCIDs. Ensure that the key pair is in PEM format. Also, upload public key to the OCI console. See To upload or paste API key in Required Keys and OCIDs.
Finger Print	Enter your finger print. See *How to get the Key's Fingerprint* in Required Keys and OCIDs.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the OCI Generative AI REST API Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection type	Select **REST API Base URL**.
Connection URL	Enter the Generative AI Service Inference API endpoint to use. For example: `https://inference.generativeai.us-chicago-1.oci.oraclecloud.com/20231130`

3. In the Security section, enter the following details:

Field	Information to Enter
Security policy	Select **OCI Signature Version 1**.

Field	Information to Enter
Tenancy OCID	Enter your tenancy OCID. See Finding Your Tenancy OCID (Oracle Cloud Identifier)
User OCID	Enter your user OCID. See Resource Identifiers.
Private Key	Enter the private key. See *How to Generate API Signing Key* in Required Keys and OCIDs. Ensure that the key pair is in PEM format. Also, upload public key to the OCI console. See To upload or paste API key in Required Keys and OCIDs.
Finger Print	Enter your finger print. See *How to get the Key's Fingerprint* in Required Keys and OCIDs.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**.

Configure the Lookup Table

Edit the values of the lookup keys as required in the recipe's lookup tables.

The recipe contains the following lookup tables:

- **AILanguageParameters**: This lookup table holds input parameters for Language AI Service.
- **GenerativeAIParameters**: This lookup table holds input parameters for Generative AI Service.

1. In the Lookups section, click the lookup name.
2. Edit the lookup table.
   - In the **AILanguageParameters** lookup table, enter required values to prepare request for `BatchDetectLanguageEntities API` in the **Oracle MaskingPIIfromEmail** integration flow.
   - In the **GenerativeAIParameters** lookup table, enter required values to prepare request for `Generative AI API` in the **Oracle GenAI Quick ReplyToEmail** integration flow.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**.

Update Integration Properties

Update integration property for the **Oracle Get Emails** integration flow.

1. In the Integrations section, click **Actions** ⋮ on the integration flow, then select **Update property values**.
2. In the Update property values panel, enter the value for the **MailSubjectToRead** integration property. For example: `Account Payment Dispute`.

   This integration property holds the value for the email subject on the basis of which emails retrieved from the Outlook server are filtered.
3. Click **Submit**.

   A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the **Oracle Get Email** integration flow, then select **Run**.
   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution. The integration now retrieves unread emails from the Outlook server and filters them based on a specific subject (for example: *Account Payment Dispute*) for further processing. Subsequently the second and third integration flows get triggered and automatic responses (without any manual interventions) are sent to emails with the specific subject.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Check your Microsoft Office 365 Outlook email inbox if you got automatic replies to emails that you had sent with the specific subject (*Account Payment Dispute*).

   Open the response and verify if sensitive information, such as bank account number, have been masked in the email.

Related Documentation

- *Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Send Compliance Documents from DocuSign to New ServiceNow Users

Use this recipe to send an IT compliance document from DocuSign to a new ServiceNow user.

Topics:

About This Recipe

This recipe sends an IT compliance document from DocuSign to the email address of a new IT service management user, when the user is created in ServiceNow. The integration uses the standard ServiceNow Adapter and the standard DocuSign Adapter.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Then you can activate and run the recipe manually. When a new user is created in ServiceNow, the integration flow gets triggered, and it sends the IT compliance document from DocuSign to the user's email address specified in the new user record form in ServiceNow. The user electronically signs the compliance document received in email and completes the process.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- ServiceNow
- DocuSign
- An account in ServiceNow with the Administrator role
- An account in DocuSign with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your ServiceNow and DocuSign instances in order to connect to these external systems using Oracle Integration and successfully send compliance documents.

Configure ServiceNow

You can use admin account credentials or custom account credentials while configuring the ServiceNow connection in Oracle Integration.

If you do not want to use an admin account, create a custom integration user and assign the required role and permissions to the user. For information about creating a custom user, see Prerequisites for Creating a Connection in *Using the ServiceNow Adapter with Oracle Integration 3*.

Configure DocuSign

You must complete the following prerequisite tasks to connect to DocuSign from Oracle Integration, and send compliance documents to new ServiceNow users.

1. Create an App and Integration Key
2. Create a Template
3. Get the Template ID

Create an App and Integration Key

You must create an app and get information such as the integration key, that is the client ID, and the client secret to successfully connect to DocuSign from Oracle Integration.

1. Log in to your DocuSign Developer account at `https://appdemo.docusign.com/home`.
2. On the DocuSign home page, click **Settings** on the title bar.
3. In the left navigation menu, scroll down to **INTEGRATIONS**, and click **Apps and Keys**.
4. On the Apps and Keys page, click **ADD APP AND INTEGRATION KEY**.
5. In the Add Integration Key dialog, enter a name for the app in the **App Name** field, and click **CREATE APP**.
6. On your app's page, under General Info, note the **Integration Key**. Click the **Copy to clipboard** icon to copy the integration key value.

   You'll need the integration key (client ID) value while configuring the DocuSign connection from Oracle Integration.
7. In the Authentication section:
   a. Under **User Application**, select **Authorization Code Grant**.
   b. Under **Secret Keys**, click **ADD SECRET KEY**
   c. Click the **Copy to clipboard** icon to copy the secret key value.
   d. Note the secret key value. You'll need it later while configuring the DocuSign connection from Oracle Integration.
8. In the Additional settings section, under **Redirect URIs**, click **ADD URI** and enter your redirect URI.

   ```
   https://{OIC_HOST}:{OIC_SSL_PORT}/icsapis/agent/oauth/callback
   ```

   Replace `OIC_HOST` and `OIC_SSL_PORT` with specific values to your Oracle Integration instance.
9. Click **Apps and Keys** to go back to the Apps and Keys page. The newly created app is displayed under **Apps and Integration Keys**.
10. Get the **API Account ID**.
    a. Under My Account Information of the Apps and Keys page, copy the **API Account ID**. Click the **Copy to clipboard** icon.
    b. Note the value of the **API Account ID**. You'll need it later while configuring the DocuSign connection from Oracle Integration.
11. Get the DocuSign **Account ID**.

a. Click the profile image on the upper-right side of the page.

b. On the menu that displays, note the Account ID that is under name or organization name. You'll need it later while configuring the DocuSign connection and also for updating the integration properties in Oracle Integration.

Create a Template

The compliance document that is sent to new ServiceNow users from DocuSign must be based on a pre-defined template. You have to create and set up this template in DocuSign.

1. Log in to your DocuSign account at https://account.docusign.com.
2. On the DocuSign home page, click **Templates** on the title bar.
3. In the left navigation pane, click **New**, and from the drop-down menu click **Create Template**.
4. In the resulting window, enter a name and description for the template.
5. In the **Add Documents** section:
   a. To add a file from your local storage, click **UPLOAD**.
   b. To add a file from one of the authorized cloud storage providers click **GET FROM CLOUD**, and choose an option from the drop-down menu. The available options are Box, DropBox, GoogleDrive, and OneDrive.

      See the topic Supported File Formats in the *DocuSign eSignature User Guide*.
6. In the **Add Recipients** section:
   a. Enter the role of the recipient in the **Role** field.
   b. Enter the name of the recipient in the **Name** field.
   c. Enter the email of the recipient in the **Email** field.
   d. To add more recipients, click **ADD RECIPIENT**, and repeat the steps above.
   e. You can also enable bulk send for multiple recipients.

      See the topic Bulk Send for Multiple Recipients in the *DocuSign eSignature User Guide*.
   f. Optionally, you can set a routing method or add an authentication method.

   To learn more, see Add Recipients in the *DocuSign eSignature User Guide*.
7. Scroll down to the **Message to All Recipients** section, and enter a standard email subject and message that you want to send to all the recipients of your template.

   See the topic Add Messages in the *DocuSign eSignature User Guide*.
8. If you don't want to add any more fields to the template, click **SAVE AND CLOSE**.
9. If you want to add fields, click **NEXT** and follow the steps described in Add Fields to Documents in the *DocuSign eSignature User Guide*.
10. Click **SAVE AND CLOSE**.

Tip: To quickly access a topic in the *DocuSign eSignature User Guide*, you can search for the topic in the DocuSign | Support page (https://support.docusign.com/en/home).

Get the Template ID

You have to get the ID of the predefined template that you created in DocuSign and pass the template ID value to the integration properties so that the right compliance document is sent to new ServiceNow users.

1. Log in to your DocuSign account at https://account.docusign.com.
2. Click **Templates** on the title bar of the home page.
3. Search for your template by entering your template's name in the Search field.
4. Click the template to open it in the detail view.
5. In the template's detail view, click the **Template ID** link.
6. In the resulting Template ID dialog, click **COPY** to copy the template ID.

   You get a message that the template ID has been copied to the clipboard.
7. Note the template ID. You'll need it later while updating the integration properties from Oracle Integration.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle DocuSign Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID (integration key) that you obtained earlier while creating an app and integration key in DocuSign. See Create an App and Integration Key.
Client Secret	Enter the client secret that you obtained earlier while creating an app and integration key in DocuSign.
Scope	Enter `signature extended`.
Instance Type	Enter `sandbox`.
Account ID	Enter the DocuSign Account ID followed by the API Account ID you obtained while creating an app and integration key in DocuSign.

3. Click **Provide Consent**.
4. In the resulting Sign in dialog, enter your Oracle Integration username and password in the **Username** and **Password** fields, and click **Sign in**.

   A browser window to grant access to DocuSign appears.
5. Enter your DocuSign account credentials, then click **Authorize** and grant access.
6. Click **Save**. If prompted, click **Save** again.

7. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

8. To return to the project workspace, click **Go back**.

Configure the Oracle ServiceNow Connection

1. In the Connections section, click the connection name.
2. In the Connection Properties section, enter the ServiceNow host name. For example: `https://instance_name.servicenow.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Username	Enter your ServiceNow username.
Password	Enter your ServiceNow password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Update the Integration Property

You must update the integration property values so that the right compliance document is sent to a new IT user in ServiceNow, and run time exception emails are sent to the right email ID.

1. In the Integrations section, click **Actions** on the **Oracle ServiceNow DocuSign User Doc SignIn** integration flow, then select **Update property values**.
2. In the Update Property Values dialog, note the three properties that have been configured - DocuSignAccountId, DocuSignTemplateId, and NotificationMail.
3. Click **DocuSignAccountId**, and in the **New Value** field, enter your DocuSign Account ID. You obtained the DocuSign Account ID while creating an app and integration key. See step 11 in Create an App and Integration Key.
4. Click **DocuSignTemplateId**, and in the **New Value** field, enter the Template ID that you obtained earlier. See Get the Template ID.
5. Click **NotificationMail**, and in the **New Value** field, enter the email ID where you want to notify in case a run time exception/error occurs while running the integration.
6. Click **Submit**.

   A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.

2. Configure the basic authentication in ServiceNow for the integration flow of the recipe.
   a. After activating the recipe, obtain the metadata URL of the integration flow.
      i. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run details**.
      ii. In the Run details panel, copy the **Metadata URL** value.
   b. Use the metadata URL to configure a basic authentication in ServiceNow. This is necessary to invoke the integration flow in Oracle Integration from ServiceNow.

      See Configure Basic Authentication in *Using the ServiceNow Adapter with Oracle Integration*.
3. Run the recipe.

   Create a new user in ServiceNow.
   a. Log in to your ServiceNow | Developer instance as a system administrator.
   b. In the navigation pane, enter `User Administration` in the Filter navigator field.

      The Self-Service menu displays all related modules.
   c. In the Self-Service menu, under **User Administration** click **Users**.
   d. On the Users page, click **New**.
   e. Fill in the required information in the new user record form.

Field	Information to Enter
User ID	Enter a unique identifier for the new user's ServiceNow log in user name.
First Name	Enter the user's first name.
Last Name	Enter the user's last name.
Department	Select the user's department from the look up list.
Title	Select the user's job title. Click the **Suggestion** icon to select from a list of suggested titles.
Email	Enter the user's email address. Note that this is the email address to which the compliance document will be sent to the user.

   f. Click **Submit**.

      The new user record appears at the top of the list in the Users page.
4. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
5. Check if you received the compliance document from DocuSign in the specified email address, and you're able to access and sign it.
   a. Sign in to the email account that you specified in the ServiceNow new user form in step 3e.
   b. Check if you received an email to review and sign the compliance document.
   c. Open the email, and click **REVIEW DOCUMENT**.
   d. Select the **I agree to use Electronic Records and Signatures** check box.
   e. Select **Continue** to access the compliance document.
   f. On the compliance document, click the **START** tag on the left. You're taken to the Signature field of the document. Optionally, drag the **START** tag to the part of the document where you want to sign.

g. Click the **SIGN** tag.

h. In the resulting Adopt Your Signature dialog, verify if your name and signature is correct, and then click **ADOPT AND SIGN**.

   If needed, you can change the information.

   - Enter your name in the **Full Name** field.
   - Enter your initials in the **Initials** field.

i. On the compliance document, click **FINISH**.

j. In the dialog that lets you know that a copy of the signed document will be sent to your email address, click **Continue**.

   You get an email with the signed compliance document.

Related Documentation

- *Using the ServiceNow Adapter with Oracle Integration 3*
- *Using the DocuSign Adapter with Oracle Integration 3*

Send Files from OCI Object Storage to Oracle ATP

Use this recipe to send files from OCI Object Storage to Oracle Autonomous Transaction Processing (Oracle ATP).

Topics:

About This Recipe

This recipe transfers files from OCI Object Storage to Oracle ATP, whenever a file is uploaded in OCI Object Storage.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a new object (file) is uploaded to a bucket in OCI Object Storage, the integration flow is triggered, and it downloads the file from OCI Object Storage. Further, the recipe sends the downloaded file to Oracle ATP as a Binary Large Object (BLOB) attachment.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462), or higher
- OCI Object Storage
- An account on OCI Object Storage with the Administrator role
- Oracle ATP
- An account on Oracle ATP with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your OCI Object Storage and Oracle ATP instances in order to successfully connect to these external systems using Oracle Integration and transfer files.

Configure Oracle Database Cloud Service

To access the Oracle Database Cloud Service using Oracle Integration, you must perform certain general configurations on your Oracle Database Cloud Service instance.

See Prerequisites for Creating a Connection in *Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*.

Also, perform the following tasks to configure Oracle Database Cloud Service for this recipe.

1. Create a schema named OBJECTSTORE_ATP. See Creating Schemas.
2. Run the following script to create a table named `FILE_TABLE` under the schema.

```
CREATE TABLE OBJECTSTORE_ATP.FILE_TABLE
        (
          FILE_NAME     VARCHAR2 (128),
          CREATION_DATE DATE,
          FILE_BLOB     BLOB
        )
```

Configure OCI Object Storage

To access OCI Object Storage instance from Oracle Integration, you must create a bucket and create a set of API signing keys in your OCI Object Storage instance. See How to use the OCI Object Storage from the Oracle Integration Cloud.

Also, ensure that the **Emit Object Events** is enabled for the bucket created.

Configure OCI Notification Service

To configure OCI events and notification services, see Configuring OCI Events and Notification Services for OCI Data Integration (OCI-DI).

Note the following while configuring notification services:

- While creating the subscription, create a HTTPS subscription. In the Create Subscription page, select HTTPS as protocol.
- In the URL field, specify your endpoint URL in the following format: `https://username:password@<host> /ic/api/integration/v2/flows/rest/project/ORCL-R-STORE_ATP_FILETRANSFER/ORACLE_OBJECTST_ATP_FILETRAN/1.0/sendcreateobjectevent`

Specify the following values while configuring events.

- In the rule conditions section, specify the following values.

Field	Information to Enter
Condition	Select **Event Type**.
Service Name	Select **Object Storage**.

Field	Information to Enter
Event Type	Select **Object – Create**.

- In the Attribute(filter) section, specify the following values.

Field	Information to Enter
compartmentName	Enter a desired compartment. This filters events from a specific compartment.
bucketName	Enter a desired Object Storage Bucket. This filters events only in the Bucket from where you want to source files for your OIC flow.
Event Type	Select **Object – Create**.

- In the Actions section, specify the following values.

Field	Information to Enter
Action Type	Select **Notifications**.
Notifications Compartment	Enter a desired compartment. This filters notifications from a specific compartment.
Topic	Select the topic created previously.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave **REST API Base URL** selected.
Connection URL	Enter the object storage endpoint that matches your location from the *API Endpoints* list. See Object Storage Service API.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **OCI Signature Version 1** selected.
Tenancy OCID	Enter your tenancy OCID. See Finding Your Tenancy OCID (Oracle Cloud Identifier).
User OCID	Enter your user OCID. See Resource Identifiers.

Field	Information to Enter
Private Key	Enter the private key. See *How to Generate API Signing Key* in Required Keys and OCIDs. Ensure that the key pair is in PEM format. Also, upload public key to the OCI console. See *To upload or paste API key* in Required Keys and OCIDs.
Finger Print	Enter your finger print. See *How to get the Key's Fingerprint* in Required Keys and OCIDs.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle DBaaS Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Oracle Wallet**.
Wallet	Enter the username of your Oracle Wallet. See Configure Oracle Database Cloud Service.
Wallet Password	Enter the password of your Oracle Wallet.
Database Service Username	Enter the username of your database service.
Database Service Password	Enter the password of your database service.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
5. To return to the project workspace, click **Go back**.

Update the Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions** on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, update the following properties:
   - **FromAddress**: This integration property holds the email address from which error notification emails are sent. Enter an email address of your choice.
   - **ToAddress**: This integration property holds the email address to which error notification emails are sent. Enter an email address of your choice.
3. Click **Submit**.

   A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe. To do so, upload an object (file) to a bucket in OCI Object Storage. See Uploading Objects to a Bucket or Folder.

   The Activity Stream pane appears displaying the status of the integration instance's execution.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Check if the file has been transferred to Oracle ATP. See List Contents of Directory in Autonomous Database.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Sync Leads Between Marketo and Salesforce

Use this recipe to synchronize leads between Marketo and Salesforce.

Topics:

About This Recipe

This is a bidirectional recipe. It creates a corresponding lead record in Marketo when a lead record is set to *Open* status in Salesforce. In addition, it creates corresponding lead records in Salesforce for Marketo leads.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe.

When a lead record's status is set to *Open* in Salesforce, the app-driven integration flow (**Oracle SFDC Marketo Lead2Lead Create**) of the recipe is triggered, and it creates a corresponding lead record in Marketo. Further, you can run the scheduled integration flow (**Oracle Marketo Salesforce Lead2Lead CreateUpdate**) of the recipe manually or specify an execution schedule for it. When triggered, this integration flow fetches the lead records from the Marketo instance and, in turn, creates or updates corresponding leads in your Salesforce instance. Basic data, such as company, country, phone number, lead status, and so on, are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 22.1.2.0.0 or higher

- Salesforce
- An account on Salesforce with the Administrator role
- Marketo
- An account on Marketo with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Marketo instances in order to successfully connect to these external systems using Oracle Integration and synchronize leads between them.

Configure Salesforce

To access Salesforce from Oracle Integration and create/read lead records, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the **Oracle SFDC Marketo Lead2Lead Create** integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce lead's status is set to *Open*. Finally, create a custom field for Salesforce lead records to store Marketo lead IDs.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

> **Note:**
>
> The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
   a. On the Salesforce Setup page:
      i. Expand **Manage Users** under the Administer section in the left navigation pane.
      ii. Click **Profiles**.
   b. On the Profiles page, click **New Profile**.
   c. On the resulting page:
      i. Select **Standard User** in the **Existing Profile** field.
      ii. Enter a name for the new profile, for example, `API Enabled`, and click **Save**.

      The new profile is now saved, and the Profile Detail page of the new profile is displayed.
   d. Click **Edit** on the Profile Detail page.
   e. On the Profile Edit page:
      i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.

ii. Scroll to the Standard Object Permissions section and perform the following actions.

- In the Accounts row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
- In the Contacts row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
- In the Price Books row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
- In the Products row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
- In the Leads row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, **Delete**, **ViewAll** and **Modify All** check boxes.

iii. Scroll to the end of the page and click **Save**.

2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

> **Note:**
>
> If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

a. On the Profile Detail page of the API Enabled profile, click **View Users**.

b. Click **New User** in the resulting page.

c. On the New User page:

i. Enter a first name and last name for the user, for example, `Integration User05`.

ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note this username.

iii. In the **User License** field, select **Salesforce**.

iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.

v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.

d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.

e. Set a password and note the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

> **Note:**
>
> If you're shown the Lighting Experience UI, switch to the Salesforce Classic UI.

f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.

   i. Stay signed in as the new user and click the user name at the top of the page to open a menu.

   ii. Click **My Settings** in the menu.

   iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.

   iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.

   v. Click the **Reset Security Token** button.
   A new security token is sent to the email address associated with the account. Note the security token.

   vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.

3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version.

4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization.

5. On your Salesforce instance, create an outbound message to trigger the **Oracle SFDC Marketo Lead2Lead Create** integration flow of this recipe.

   a. On the Salesforce Setup page:

      i. Scroll to the Build section in the left navigation pane.

      ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.
      If you're shown the Understanding Workflow page, click **Continue**.

   b. Click **New Outbound Message** in the resulting page.

   c. On the New Outbound Message page:

      i. Select **Lead** in the **Object** field, and click **Next**.

      ii. Enter the following details for the new outbound message:

      - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
      - Optionally, enter a description for the message.
      - In the **Endpoint URL** field, enter the following URL:

        ```
        https://<OIC_FQDN:OIC_SSL_PORT/ic/ws/integration/v2/flows/
        salesforce/project/ORCL-R-SFDC_MARKT_SYNC_LEADS/ORCL-R-
        SFDC_LEA_SYNC_TO_MKT_LEA/1.0
        ```

        Where `OIC_FQDN` and `OIC_SSL_PORT` are the fully-qualified domain name and port of your Oracle Integration instance (for example, `oicinstance.example.com:443`).

      - In the **User to send as** field, select the user account using which you want to send the message.
      - Under **Lead fields to send**, select the required fields and click **Add**.
      - Click **Save**.

The new outbound message is created and displayed on your browser.

6. Now, create a workflow rule to send the outbound message (created previously) when a Salesforce lead's status is set to *Open*.
   a. On the Salesforce Setup page:
      i. Scroll to the Build section in the left navigation pane.
      ii. Expand **Create**, then **Workflow & Approvals**, and then click **Workflow Rules**.

      If you're shown the Understanding Workflow page, click **Continue**.
   b. Click **New Rule** in the resulting page.
   c. On the New Workflow Rule page:
      i. Select **Lead** in the **Object** field, and click **Next**.
      ii. Enter the following details for the new workflow rule:
         - Enter the rule's name.
         - Optionally, enter a description for the rule.
         - In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
         - In the Rule Criteria section, select the **formula evaluates to true** option, and define the following rule criteria:

           ```
           (Lead: Lead Status EQUALS Open - Not Contacted) AND (Lead:
           Company NOT EQUAL TO null)
           ```

         - Click **Save & Next**.
      iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
      iv. On the Select Existing Actions page:
         - Select **Outbound Message** in the **Search** field.
         - In the **Available Actions** box, select the outbound message you created previously, for example, `Outbound Message: <name of your outbound message>`.
         - Click **Add** to move your selection to the **Selected Actions** box.
         - Click **Save**.
      v. Click **Done**.

      The new workflow rule is created and displayed on your browser.
   d. Click **Activate** to activate the rule.
7. Create a custom field for lead records.

   This recipe uses unique IDs associated with lead records in Marketo to synchronize Marketo leads with Salesforce leads. Create a custom field for Salesforce lead records to hold the Marketo IDs.
   a. On the Salesforce Setup page:
      i. Scroll to the Build section in the left navigation pane.
      ii. Expand **Customize**, then **Leads**, and then click **Fields**.

b. On the Lead Fields page, scroll to the Lead Custom Fields & Relationships section and click New.

c. On the New Custom Field page:

   i. Find and select the **Text** radio button, and click **Next**.

   ii. Enter the following details for the new custom field:

   - Enter `Lead Extension ID` as the field label. Note that the field name is automatically populated based on the label you enter.
   - Enter `50` as the length.
   - Optionally, enter a description for the new field.
   - Select the **External ID** check box.
   - Click **Next**.

   iii. On the Establish field-level security page, select the **Visible** check box in the header row to grant edit access to the new field for all profiles. Click **Next**.

   iv. Leave all the layout check boxes selected and click **Save**.

On the Lead Fields page, you can see the new field added under the Lead Custom Fields & Relationships section.

Configure Marketo

To access Marketo from Oracle Integration and create/read lead records, you'll need to perform the following configuration tasks on your Marketo instance.

1. Create an API-only user role.
2. Create a customer service.
3. Obtain the Client ID and Client Secret.
4. Obtain the Munchkin ID.

For detailed information on the above steps, see Prerequisites for Creating a Connection in *Using the Marketo Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select **Production** or **Sandbox** based on your Salesforce instance type.

Field	Information to Enter
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Salesforce Username Password Policy** selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce.
Password	Enter the password of the account created for Oracle Integration on Salesforce. **Note:** To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle Marketo Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Munchkin ID that you obtained earlier. See Configure Marketo.
3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID obtained earlier. See Configure Marketo.
Client Secret	Enter the client secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Lookup Table

The **ORACLE-BRT-MKT_SFDC_LOOKUP** table contains a sample mapping of lead-status values. Edit the table to map the lead-status values in your Marketo instance to that of your Salesforce instance.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

Key	Description
Marketo_Value	Enter the lead-status values available in your Marketo instance (in separate rows). For example, `Open`, `Qualified`, and so on.
Salesforce_Value	Enter a corresponding lead-status value available in your Salesforce instance against each Marketo value.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration property for the **Oracle SFDC Marketo Lead2Lead Create** integration flow.

1. In the Integrations section, click **Actions** on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the following integration property.
   - **EmailTo**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
3. Click **Submit**.

   A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe to create a Marketo lead when a Salesforce lead's status is set to *Open*.
   a. Log in to your Salesforce instance.
   b. On the Salesforce Setup page (in the Classic UI), click the **Leads** tab.
   c. On the Leads Home page, in the **View** field, select an option to list the lead records, and click **Go!** if necessary.

      The leads on your Salesforce instance are displayed.
   d. Search for a lead record that is not in the *Open* state, and click its name to view it.
   e. On the lead's page, click **Edit**.
   f. On the Lead Edit page:
      i. In the **Lead Status** field, select **Open - Not Contacted**.
      ii. Click **Save**.

   You've now successfully triggered the recipe, which creates a corresponding lead record in Marketo.

3. Run the recipe to create leads in Salesforce for Marketo lead records.

    a. In the Integrations section of the project workspace, click **Actions** ⋮ on the **Oracle Marketo Salesforce Lead2Lead CreateUpdate** integration flow, then select **Run**.

    The Configure and run page is displayed with the following parameter. Update the parameter with appropriate value.

    - `LastRun`: This parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Marketo lead records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the lead records from the beginning, set the value as zero.

    b. On the Configure and run page, click **Run**.

    You've now successfully submitted the integration for execution.

    > **Note:**
    >
    > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

4. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
5. Log in to your Marketo instance and check for the new leads created.
    a. On the landing page, click the **My Marketo** tab.
    b. In the left navigation pane, click **All People (APA)**.
    c. Under the **People** tab, search for the required lead records.

    You can search for the leads by their names, IDs, or any other configured fields.
6. Log in to your Salesforce instance and check for the new leads created.
    a. On the Salesforce Setup page (in the Classic UI), click the **Leads** tab.
    b. On the Leads Home page, in the **View** field, select an option to list the lead records, and click **Go!** if necessary.

    The list of leads is displayed, which contains the records imported from Marketo.

    c. To view a lead record, click its name.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Marketo Adapter with Oracle Integration 3*

Sync Oracle NetSuite Customers with Oracle CX Sales and B2B Service Accounts

Use this recipe to synchronize Oracle NetSuite customers with Oracle CX Sales and B2B Service accounts.

Topics:

About This Recipe

This recipe creates or updates a corresponding Oracle CX Sales and B2B Service account when a customer is created or updated in Oracle NetSuite. Additionally, whenever an account is created or updated in Oracle CX Sales and B2B Service, the recipe creates or updates a corresponding customer in Oracle NetSuite.

To use the recipe, you must install the recipe and configure the connection and other resources within it. The recipe contains three integration flows - **Oracle NetSuite CX Sales Customer Upsert** , **Oracle CX Sales NetSuite Customer Create** and **Oracle CX Sales NetSuite Customer Update**.

- When a new customer is created or a customer record is updated in Oracle NetSuite, the user event script associated with the customer gets triggered in NetSuite. The user event script provides the JSON data of the customer and subsequently triggers the **Oracle NetSuite CX Sales Customer Upsert** integration flow in Oracle Integration. The integration then creates or updates a corresponding account record in Oracle CX Sales and B2B Service.
- When a new account is created in Oracle CX Sales and B2B Service, the **Oracle CX Sales NetSuite Customer Create** integration flow gets triggered in Oracle Integration. The integration creates a corresponding customer record in Oracle NetSuite.
- When an account is updated in Oracle CX Sales and B2B Service, the **Oracle CX Sales NetSuite Customer Update** integration flow gets triggered in Oracle Integration. The integration then updates the corresponding customer record in Oracle NetSuite.

Basic data such as customer/account name, phone, email, company name and address are synchronized between Oracle NetSuite and Oracle CX Sales and B2B Service platforms.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- Oracle NetSuite
- An account on Oracle NetSuite with the Administrator role
- Oracle CX Sales and B2B Service
- An account on Oracle CX Sales and B2B Service with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite and Oracle CX Sales and B2B Service instances in order to successfully connect to these external systems using Oracle Integration, and synchronize customers and accounts.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL.
2. Perform the TBA-related configuration tasks. See Prerequisites for the Token-Based Authentication Security Policy.

   Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
List	Currency	Full
List	Customers	Full
List	Subsidiaries	Full
List	Contacts	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Create a custom field for customer record. See Creating Custom Transaction Body Fields. While creating a custom field, enter the following values.
   a. Specify the value for **Label** as `CxSalesPartyNumber .`
   b. Specify the value for **ID** as `id_cxsalespartynumber`.
   c. In the **Applies To** tab, select **Customer**.
   d. Click **Save**.
4. Configure and deploy the User Event script.
   a. Download the sample user event script: NS-CXSales.js.
   b. Open the script in a text editor such as Notepad++ and make the following updates:
      i. On **line 30**, there's a variable with the value set to `***your admin user email***`. Replace this value with the email address of the integration user that you created. The end result should look like this: `var integrationUserEmail = "john@netsuite.com"`.
      ii. On **line 85**, there's a variable with the value set to `*The base URL of your Oracle Integration instance*`. Replace this value with the domain name of your Oracle Integration instance. Make sure to omit everything after the `.com` top-level domain.

iii. On **line 83**, there's a variable with the value set to your username. Replace this value with the username of your Oracle Integration instance.

iv. On **line 84**, there's a variable with the value set to your password. Replace this value with the password of your Oracle Integration instance.

v. Save the changes made to the script.

c. In NetSuite, navigate to **Documents**, then **Files** and then **SuiteScripts**.

d. Within the SuiteScripts folder, create a new folder. For example: `Stripe Integration`.

e. Upload the script that you modified and saved in step b to the newly created folder.

f. Go to **Customization**, then **Scripting**, and then **Scripts**.

g. In the **Script File** field, start typing `Oracle_NS_Stripe_Customer_Create.js`, and select the auto-completed option.

h. Click **Create Script Record**.

i. Set the **Name** field to `Oracle NS Stripe Customer Script` and **ID** to `_oracle_stripe_cust`.

j. Click **Save** and then click **Deploy Script**.

k. Set the **Applies To** field to `Customer` and the **ID** to `_oracle_stripe_cust_depl`.

l. Set the **Status** to **Deployed**.

m. In the **Audience** tab, select all check boxes where applicable or select all items in the respective lists.

n. Click **Save**.

Configure Oracle CX Sales and B2B Service

To successfully connect to Oracle CX Sales and B2B Service using Oracle Integration, you must perform certain configurations on your Oracle CX Sales and B2B Service instance.

1. Create an integration user account.

   To invoke an Oracle CX Sales and B2B Service service catalog or event catalog web service from , you create a separate user.

   a. Log in to the Oracle CX Sales and B2B Service with a user with system administrator privileges.

   b. Go to **Navigator** > **My Team** > **Manage Users**.

   c. In the Manage Users page, click **Manage Users**.

   d. Click the **Create New User** icon beside **Show Photo**.

   e. Enter the following information, and click **Save**.

Field	Description
Last Name	Enter `FUSION_APPS_ICS_APPID`
Email	Enter a valid email address.
Hire Date	Enter the date.
User Name	Enter `FUSION_APPS_ICS_APPID`.
Person Type	Enter `Employee`.
Legal Employer	Select a valid legal organization.

Field	Description
Business Unit	Select a valid business unit.
Send user name and password	Select this checkbox.

A notification email is sent to the email address after the user is created.

f. Log out of the Oracle CX Sales and B2B Service.

g. Log in to the Oracle CX Sales and B2B Service instance with FUSION_APPS_ICS_APPID and the temporary password provided in the notification email.

h. Change the password when prompted at the first log in.

The Oracle CX Sales and B2B Service welcome page appears.

i. Log out of the Oracle CX Sales and B2B Service.

2. Assign integration roles.

Use the Oracle Security Console to assign the integration user with certain roles and privileges.

> **Note:**
>
> Access to the Security Console is provided by the predefined **Security Manager** role.

a. Select **Navigator** > **Tools** > **Security Console**.

b. On the top right corner of the window, click **Create Role**.

c. In the Create Role: Basic Information page, create a new record with the following information and click **Next**.

Parameter	Value
Role Name	OIC Integration Role
Role Code	INT_ OIC_ Integration_ Role
Role Category	CRM - Job Roles
Description	Custom Role for Accessing OSC Services Catalog

d. In the Create Role: Functional Security Policies page, click **Add Functional Security Policy**.

e. In the Add Function Security Policy page, enter `FND_MANAGE_CATALOG_SERVICE_PRIV` in the Search box and click **Add Privilege to Role**. Click **Next**

f. In the Create Role: Data Security Policies page, click **Next**.

g. Add the **Sales Administrator** and the **SOA operator** roles, and then click **Next**.

To add the **Sales Administrator** role, do the following on the Create Role: Role Hierarchy page:

i. Click **Add role**.

ii. In the Search field, enter `Sales Admin`.

iii. Select the **Sales Administrator** role, and then click **Add Role Membership**.

iv. Close the Add Role Membership window.

To add the **SOA Operator** role, do the following on the Create Role: Role Hierarchy page:

i. Click **Add role**.

ii. In the Search field, enter `SOA Operator`.

iii. Select the **SOA Operator** role, and then click **Add Role Membership**.

iv. Close the Add Role Membership window.

h. Assign the integration user to the roles.

On the Create Role: Users page, click **Add user** and then do the following:

i. In the Search field, enter `FUSION_APPS_ICS_APPID`.

ii. Choose the **FUSION_APPS_ICS_APPID** user and then click **Add user to Role**.

iii. Close the Add user window.

i. Click **Next**, review the details on the Summary and Impact page, and click **Save and Close**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle NetSuite WSDL URL. For example: `https://webservices.netsuite.com/wsdl/v2022_1_0/netsuite.wsdl`. See Configure Oracle NetSuite.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Token Based Authentication**.
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite.
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Configure the Oracle CX Sales and B2B Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the username of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Configure Oracle CX Sales and B2B Service.
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Lookup Table

Edit the **ORACLE-BRT-EC_NS_SETTINGS** lookup table.

1. In the Lookups section, click the lookup name.
2. Edit lookup keys in the **ECCode** and **NSCode** columns and map Oracle NetSuite values with Oracle CX Sales and B2B Service values.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. Log in to your Oracle NetSuite instance and create a new customer or update a customer.

      This triggers the **Oracle NetSuite CX Sales Customer Upsert** integration flow which subsequently creates or updates a corresponding account in Oracle CX Sales and B2B Service.

b. Log in to your Oracle CX Sales and B2B Service instance and create a new account.

   This triggers the **Oracle CX Sales NetSuite Customer Create** integration flow which creates a corresponding customer in Oracle NetSuite.

c. Now, update an account in your Oracle CX Sales and B2B Service instance.

   This triggers the **Oracle CX Sales NetSuite Customer Update** integration flow which updates a corresponding customer in Oracle NetSuite.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Verify if a corresponding account has been created or updated in your Oracle CX Sales and B2B Service instance for the new/updated Oracle NetSuite customer.
5. Similarly, verify if a corresponding customer has been created or updated in your Oracle NetSuite instance for the new/updated Oracle CX Sales and B2B Service account.

Related Documentation

- *Using the Oracle NetSuite Adapter with Oracle Integration 3*
- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*

Sync Oracle NetSuite Items with QuickBooks Products

Use this recipe to synchronize Oracle NetSuite items with products in QuickBooks.

Topics:

About This Recipe

This recipe synchronizes inventory and sales service items in Oracle NetSuite with products in QuickBooks, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe package and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

The recipe has the following scheduled integration flows:

- **Oracle NS ServiceItem QB Product Sync**: When triggered, the integration flow fetches the sales service items from Oracle NetSuite and creates corresponding products in QuickBooks.
- **Oracle NSInventoryItem QBProduct Sync**: When triggered, the integration flow fetches the inventory items from Oracle NetSuite and creates corresponding products in QuickBooks.

Subsequently, the recipe updates the Oracle NetSuite item record with the QuickBooks product ID in the `Quickbook ProductItem Id` field. If the recipe fails to do so, the corresponding QuickBooks product is marked as inactive. Basic data, such as account and unit price are synchronized between the two platforms.

> **Note:**
>
> The recipe creates products in QuickBooks only for Oracle NetSuite item records in which the `Quickbook ProductItem Id` field is empty.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- Oracle NetSuite
- An account on Oracle NetSuite with the Administrator role
- QuickBooks
- An account on QuickBooks with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite and QuickBooks instances in order to successfully connect to these external systems using Oracle Integration and synchronize items and products.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

In addition, you must create a NetSuite saved search. The search criterion specified in the saved search will be used to fetch the sales orders that are to be imported to Oracle ADW.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL.
2. Perform the TBA-related configuration tasks. See Prerequisites for the Token-Based Authentication Security Policy.

   Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
List	Currency	Full
List	Customers	Full
List	Subsidiaries	Full
List	Contacts	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Configure saved search to get the inventory items that satisfy the search criterion. The items that are the output of the search result will be created as products in QuickBooks.

a. Navigate to the New Saved Search window using any one of the following navigation options:
   - On the NetSuite home page, select **Reports**, then choose **Saved Searches**, then **All Saved Searches**, and then **New**.
   - On the NetSuite home page, select **Transactions**, then choose **Management**, then **Saved Searches**, and then **New**.

b. On the New Saved Search window, select the object type to perform the search operation. For example, select **Item**.

c. Add a title and ID for the saved search in the SEARCH TITLE and ID fields respectively.

   The title and ID for the saved search will help you in keeping track of the saved search when you want to reuse it to re-query latest data that matches the search criteria.

d. Set the search criteria conditions. Select the **Criteria** tab and specify the conditions. For example, you can specify a condition to filter items with the custom field **Quickbook ProductItem Id** as empty.If the **USE EXPRESSIONS** check box is selected, you can also use additional options such as AND, OR, NOT. Also, specify the type to be **Inventory**.

e. Configure the search results that are to be displayed. Select the **Results** tab, and in the **Columns** tab of the page, select the fields that you want to be displayed in the search result.

Similarly, create another saved search for sales service items. Specify a condition to filter items with the custom field **Quickbook ProductItem Id** as empty, type as **Service Item**, and the subtype as **Sales**.

Configure QuickBooks

To access the QuickBooks instance from Oracle Integration and synchronize items and products, you'll need to create a user account for Oracle Integration on QuickBooks and make a few other configurations. See Prerequisites for Creating a Connection in *Using the QuickBooks Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle NetSuite Connection

Use the following steps to configure the **Oracle NetSuite** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, `https://webservices.netsuite.com/wsdl/<NetSuite_application_version>/netsuite.wsdl`, where *<NetSuite_application_version>* is the version of the NetSuite application. For example, v2015_1_0. See Assemble the Oracle NetSuite WSDL URL in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Token Based Authentication**.
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Create an Integration Record for Oracle Integration in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See Create an Access Token for the User Account in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier. See Make a Note of the NetSuite Account ID in *Using the Oracle NetSuite Adapter with Oracle Integration 3*. **Note:** You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle QuickBooks Connection

Use the following steps to configure the **Oracle QuickBooks** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
QuickBooks Environment	Select **Production**.

3. In the Security section, enter the following details:

Field	Information to Enter
Client id	Enter the Client ID obtained when you configured QuickBooks. See Configure QuickBooks.
Client secret	Enter the Client Secret obtained when you configured QuickBooks.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

   You'll now be redirected to the QuickBooks sign in page.
7. Enter your QuickBooks account credentials.

   A page is displayed asking for permission to interact with the account.
8. Click **Yes**.

   You're informed that access is allowed. You can now switch back to the Oracle QuickBooks Connection window of Oracle Integration to test your connection.
9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
10. To return to the project workspace, click **Go back**  .

Configure the Lookup Table

Edit the **Account_lookup** lookup table to map the account names in Oracle NetSuite with QuickBooks account IDs.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

netsuite_account_name	quickbooks_account_id
1150 Employee Advances	57
Inventory Asset	81
Accounts Receivable	84
1090 Undeposited Funds	4
Uncategorized Income	30
1300 Prepaid Expenses	3
6010 Advertising	7
6020 Automobile Expense	55
4000 Sales	79
Cost of Goods Sold	80
6080 Equipment Rental	29
6100 Insurance Expense	11
6150 Office Expense	15

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
      i. To synchronize sales service items, select **Oracle NS ServiceItem QB Product Sync**.
      ii. To synchronize inventory items, select **Oracle NSInventoryItem QBProduct Sync**.
   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. In your QuickBooks instance, check for new products.
   a. Log in to your QuickBooks instance.
   b. Click **Products and Services** in the homepage, and check for new products.

Related Documentation

- *Using the Oracle NetSuite Adapter with Oracle Integration 3*
- *Using the QuickBooks Adapter with Oracle Integration 3*

Sync Workday Employees with ServiceNow Users

Use this recipe to synchronize employees in Workday with ServiceNow users.

Topics:

About This Recipe

This recipe synchronizes Workday employees with ServiceNow users on a scheduled basis. It creates or modifies user records in ServiceNow according to the employee data received from Workday. In addition, if a ServiceNow user creation fails, the recipe creates a corresponding ticket in ServiceNow.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

The scheduled integration **Oracle Workday ServiceNow Employee Sync** reads the employee records from Workday and creates or modifies corresponding users in ServiceNow. Subsequently, if any ServiceNow user creation fails, the scheduled integration **Oracle Workday ServiceNow Emp Onboarding Ticket** reads the failed records and creates or updates corresponding tickets in ServiceNow. The app-driven integration **Oracle REST Workday ServiceNow** is triggered by each of the scheduled integration flows for their respective tasks.

Basic employee data such as first name, last name, email, and so on are synchronized between Workday and ServiceNow.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Workday
- An account on Workday with the Administrator role
- ServiceNow
- An account on ServiceNow with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Workday and ServiceNow instances in order to successfully connect to these external systems using Oracle Integration and synchronize employees and users.

Configure Workday

To access Workday using Oracle Integration and sync employees, you must perform certain configuration tasks on your Workday instance.

For general configuration tasks, see Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.

In addition, you must create an Enterprise Interface Builder (EIB) report to import and export Workday data in bulk. And, before you create an EIB report, you must create a custom report with custom fields, calculated fields, and business objects.

1. Create custom fields.
    a. In Workday, obtain access to **Create Custom Object**.
    b. Select **Workday Object** as **Worker (Effective Dated)** and **Custom object Name** as **SN_TicketId**.
    c. Select the values as below.
        - **General Settings**

Custom Object Name	Custom Object Name	**SN_TicketId**
Web Service Alias	ID	snTicketId
Allow Multiple Instances per Object?		No

- **Field Definitions**

Current Fields	Field Label	**SN_TicketId**
	Field Type	Text

- **Permissions**

Security	Domain	Person Data: Personal Data, Person Data: Personal Information

- **Edit Custom Fields**

Edit Custom Fields	Display Value	**SN_TicketId**

d. In Workday, search for **Edit Worker additional data business process**.

e. In the **Edit worker additional data business process**, go to **Action** and click **Edit definition**.

f. Add a new step and select values as below.

> **Note:**
>
> In **Group** column, select the **Security Group** which should be allowed to edit the custom field.

Step	Order	Type	Specify	Optional	Group	All	Run as User	Due Date	Due Data Is Based On Effective Date	Complete
	a5	Edit Additional Data	**SN_TicketId**	No	HR Administrator					

g. Click **OK**.

Similarly, create custom fields to store **Ticket_ErrorMessage**, **User_ErrorMessage** and **ServiceNowUserId**.

> **Note:**
>
> Provide **Get access of Person Data: Personal Data, Person Data: Personal Information to ISU user** being used for getting data from custom report.

2. Create calculated fields.
   a. In the quick find box, search for **Create Calculated Fields** task.
   b. In the pop-up window, enter the **Field name**. For example: `CF_new_hire_or_updated`. Also, select the **Business Object** as **Worker**, and **Function** as **Evaluate Expression**.
   c. Click **OK**. You are redirected to **Create Calculated Field - Evaluate Expression** page.

d. Under Calculation, select the **Field Type** as **Text** and **Default Value** as **Empty String**.

e. In the Condition column, create the following calculated fields:

- **CF-T/F_oic_updated**

And/Or	(Field	Calculated Field Details	Operator	Comparison Type	Comparison Value	)
And	**CF_Legal_name_change**	**Business Object**: Worker **Source Field**: Worker History **Related Business Object**: Action Event **Condition**: CF_legal_name_change? **Sort Field**: Effective Date **Sort Direction**: Descending (Z to A) **Instance to be Returned**: First Occurrence **Function**: True or False	is not empty	NA	NA	
Or	**CF_new_event_exist**	**Business Object**: Worker **Source Field**: Worker History **Related Business Object**: Action Event **Condition**: CF_effective date range **Fields To Aggregate**: Business Process Event **Function**: True or False	is not empty	NA	NA	
Or	(**CF_usererrormsg**	**Business Object**: Worker **Lookup Field**: Worker Additional Data **Related Business Object**: Worker Custom Data Snapshot **Return Value**: User_ErrorMessage **Function**: True or False	equal to	Value specified in this filter	User failed	
And	**CF_user_ticket_error**	**Business Object**: Worker **Lookup Field**: Worker Additional Data **Related Business Object**: Worker Custom Data Snapshot **Return Value**: Ticket_ErrorMessage **Function**: True or False	equal to	Value specified in this filter	Ticket failed	)

- **CF_legal_name_change?**

And/Or	(Field	Calculated Fields Details	Operator	Comparison Type	Comparison Value	)
And	**Business Process Type**	**Business Object**: Event **Function**: True or False	in the selection list	Value specified in this filter	Legal Name Change	
And	**Status**	**Business Object**: Event **Function**: True or False	greater than or equal to	Value specified in this filter	Successfully Completed	
And	**Effective Date**	**Business Object**: Action Event **Function**: True or False	greater than or equal to	Value from another field	Prompt-Start Date	

- **CF_oic_is_new_hired?**

And/Or	(Field	Calculated Fields Details	Operator	Comparison Type	Comparison Value	)
And	**Hire Date**	NA	greater than or equal to	Value from another field	Prompt-Start Date	
And	**CF_Legal_name_change**	**Business Object**: Worker **Source Field**: Worker History **Related Business Object**: Action Event **Condition**: CF_legal_name_change ? **Sort Field**: Effective Date **Sort Direction**: Descending (Z to A) **Instance to be Returned**: First Occurrence **Function**: True or False	is empty	NA	NA	
And	**CF_new_event_exist**	**Business Object**: Worker **Source Field**: Worker History **Related Business Object**: Action Event **Condition**: CF_effective date range **Fields To Aggregate**: Business Process Event **Function**: True or False	is empty	NA	NA	

f. In the **Return Value If Condition is True** column, create the following calculated fields.

Calculated Field	Details
CF_oic_updated	**Business Object:** Worker **Text Constant**: Updated

Calculated Field	Details
CF_oiC_NEW_HIRED	**Business Object**: Worker **Text Constant**: NEW HIRE

Configuration details for other related calculated fields are listed below.

Calculated Field	Details
CF_legal_name_change	**Business Object**: Worker **Source Field**: Worker History **Related Business Object**: Action Event **Condition**: CF_legal_name_change? **Sort Field**: Effective Date **Sort Direction**: Descending (Z to A) **Instance to be Returned**: First occurrence **Function**: True or False
CF_Ticket_ErrorMessage?	**Business Object**: Worker **Function**: Lookup Related Value **Lookup Field**: Worker Additional Data **Related Business Object**: Worker Custom Data Snapshot **Return Value**: Ticket_ErrorMessage
CF_work_email	**Business Object**: Worker **Function**: Lookup Value As Of Date **Source Field**: Email – Work **Data As Of**: Effective Date: empty **Data Entry Date**: Entry Date: CF_Last_Functionally_updated_-1millisecond
CF_Last_Functionally_updated_-1mili second	**Business Object**: Global **Function**: Increment or Decrement Date **Date Field**: Last Functionally Updated **Milliseconds to Add or Subtract**: -1 **Return Blank Date on Error**: Yes
CF_proposed_Supervisory Organization	**Business Object**: Worker **Function**: Lookup Value As Of Date **Source Field**: Supervisory Organization **Data As Of**: Effective Date: empty **Data Entry Date**: Entry Date: CF_Last_Functionally_updated_-1mili second
CF_proposed_Phone - Primary Work	**Business Object**: Worker **Function**: Lookup Value As Of Date **Date Source Field**: Phone - Primary Work **Data As Of**: Effective Date: empty **Data Entry Date**: Entry **Date**:CF_Last_Functionally_updated_-1mili second

Calculated Field	Details
CF_User_ErrorMessage?	**Business Object**: Worker **Function**: Lookup Related Value **Lookup Field**: Worker Additional Data **Related Business Object**: Worker Custom Data Snapshot **Return Value**: User_ErrorMessage
CF_previous_business_title	**Business Object**: Worker **Function**: Lookup Value As Of Date **Source Field**: Business Title **Data As Of**: Effective Date: empty **Data Entry Date**: Entry Date:CF_Last_Functionally_updated_-1mili second

Also, configure the following calculated fields with the following conditions.

- **CF_effective date range**

And/Or	(Field	Calculated Field Details	Operator	Comparison Type	Comparison Value	)
And	**Business Process Definition**	**Business Object**: Action Event **Function**: True or False	in the selection list	Value specified in this filter	• Change Job (Default Definition) • Change Job for Global Modern Services • Change Job for IT Services Group • Change Job for Sales • Change Organization Assignments for Worker (Default Definition) • Change Organization Assignments for Worker for Field Sales - North America Group • Change Organization Assignments for Worker for Global Modern Services • Change Organization Assignments for Worker for Marketing • Change Organization Assignments for Workers by Organization (Default Definition) • Change Organization Assignments for Workers by Organization for Global Modern Services • Contact Change (Default Definition) • Edit Worker Additional Data Event (Default Definition) • Edit Worker Additional Data Event for Global Modern Services • Move Worker (Supervisory) (Default Definition) • Move Worker (Supervisory) for Global Modern Services • Move Workers (By Organization) (Default Definition) • Move Workers (By Organization) for Global Modern Services	

And/Or	(Field	Calculated Field Details	Operator	Comparison Type	Comparison Value	)
					• Move Workers (Supervisory) (Default Definition) • Move Workers (Supervisory) for Global Modern Services • Personal Information Change (Default Definition) • Personal Information Change for Global Modern Services • Switch Primary Job (Default Definition) • Switch Primary Job for Global Modern Services • Title Change (Default Definition) • Title Change for Global Modern Services • Work Contact Change (Default Definition)	
And	**Status**	**Business Object**: Action Event **Function**: True or False	equal to	Value specified in this filter	Successfully Completed	
And	**Effective Date**	**Business Object**: Action Event **Function**: True or False	greater than or equal to	Value from another field	Prompt - Start Date	

3. Create a custom report.
   a. Log in to the Workday instance.
   b. In the quick find box, search for **Create custom report**.
   c. In the Create Custom Report window:
      i. Enter the **Report Name**. For example: `RPT_new_hire_report`.
      ii. Select the **Report Type** as **Advanced**.
      iii. Check the **Enable As Web Service** box.
      iv. Uncheck the **Optimized for Performance** box.
      v. Select the **Data Source** as **Active and Terminated Workers**.
      vi. Click **OK**. You are redirected to Edit Custom Report page.
      vii. Under the **Columns** tab, add the **Fields** and **Custom Field**s required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

Business Object	Fields	Column Heading Override XML Alias
Worker	**Employee ID**	Employee_ID
Worker	**First Name**	firstName
Worker	**Last Name**	lastName
Worker	**Email - Work**	Email_-_Work
Worker	**Hire Date**	Hire_Date
Worker	**Worker Type**	Worker_Type
Worker	**Business Title**	businessTitle
Worker	**Supervisory Organization**	Supervisory_Organization

> **Note:**
>
> Choose **Column Heading Override** as **Department**.

Worker	**Phone - Primary Work**	primaryWorkPhone
Worker	**CF_new_hire_or_updated**	CF_new_hire_or_updated
Worker Additional Data	**ServiceNowUserId**	servicenowuserid
Worker Additional Data	**SN_TicketId**	snTicketid
Worker Additional Data	**User_ErrorMessage**	userErrormessage
Worker Additional Data	**Ticket_ErrorMessage**	ticketErrormessage

d. Click **OK**.

e. In Group column heading section, select all business object as below. (Group Column Heading for each business object will be blank)

Group Column Headings

Business Object	Group Column Heading	Group Column Heading XML Alias
Worker Additional Data		Worker_Additional_Data_group

f. After creating custom reports, you can specify how the report results should be sorted.

g. Under the **Filter** tab, add the **Fields** and **Custom Fields** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

And/Or	(Fields	Operator	Comparison Type	Comparison Value	)
And	**Terminated (based on report date)**	equal to	Value specified in this filter	NA	
And	(**Hire Date**	greater than or equal to	Prompt the user for the value	Default Prompt	
Or	**CF_Legal_name_change**	is not empty	NA	NA	
Or	(**CF_new_event_exist**	is not empty	NA	NA	

And/Or	(Fields	Operator	Comparison Type	Comparison Value	)
And	(**Email - Work**	none in the selection list	Value from another field	CF_work_email	
Or	**CF_Ticket_Error Message?**	equal to	Value specified in this filter	ticket failed	
Or	**CF_User_Error Message?**	equal to	Value specified in this filter	user failed	
Or	**Business Title**	not equal to	Value from another field	CF_previous_business_title	
Or	**Supervisory Organization**	not in the selection list	Value from another field	CF_proposed_Supervisory Organization	
Or	**Phone - Primary Work**	not equal to	Value from another field	CF_proposed_Phone - Primary Work	)))

h. In the Prompts section, check the **Populate Undefined Prompt Defaults** box.

> **Note:**
>
> The data source **Active and Terminated Workers** uses built-in prompts such as **Start Date**, **End Date**, and so on. Based on the report you have chosen, Workday prompts you to fill in specific reporting criteria.

4. Create an EIB report.
   a. Log in to the Workday instance.
   b. In the quick find box, search for **Create EIB** task.
   c. Enter the **Name** of the EIB task (**EIB_new_hire_report**).
   d. In the **Create EIB** pop-up, choose **Outbound** and click **OK**.
   e. In the **General Settings**, enter comments and then click **Next**.
   f. Under the **Get Data** tab, select the **Data Source Type** as **Custom Report**.
   g. Select the **Custom Report** as **RPT_new_hire_report** and then click **Next**.
   h. Under the **Transform** tab, select the **Transformation Type** as **New Custom Report Transformation**, then click **Next**.
   i. Under the **Deliver** tab, perform the following steps.
      i. Select the **Delivery Method** as **Workday Attachment**.
      ii. Enter the **File Name** with the relevant file extension. For example, `output.csv`.
      iii. Specify the **Document Retention Policy (in Days)** as **should be less than 180**, and then click **Next**.
      iv. In the **Details** section, select the **MIME Type** as **CSV**.
      v. Click **Next**.
   j. Under the **Review and Submit** tab, verify the details, then click **OK**.
   k. Once you create the EIB report, the **View Integration System** window appears. In the top-left corner, click the ellipsis next to the EIB task name, choose **Enterprise Interface**, and then select **Configure Transformation**.

l. In the **Configure Transformation** window, update the report transformation details, click **OK**, and then click **Done**.

m. In the **View Integration System** window, in the top-left corner, click the ellipsis next to the EIB task name, choose **Integration**, and then select **Launch/Schedule**.

n. In the **Launch/Schedule Integration** pop-up, select the **Run Frequency** as **Run Now**, and then click **OK**.

o. In the **Schedule an Integration** window, specify the **Start Date** and **Hire Date**, and then click **OK**.

The EIB report gets generated.

Configure ServiceNow

You can use admin account credentials or custom account credentials while configuring the ServiceNow connection in Oracle Integration.

If you do not want to use an admin account, create a custom integration user and assign the required role and permissions to the user. For information about creating a custom user, see Prerequisites for Creating a Connection in *Using the ServiceNow Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select **REST API Base URL**.
Connection URL	Enter your Oracle Integration instance URL. For instance, `https://<instance-name>.com/`.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Basic Authentication**.
Username	Enter the username of your Oracle Integration account.
Password	Enter the password of your Oracle Integration account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Configure the Oracle Workday Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Hostname	Enter the Workday host name. For example: `https://wd2-impl-services1.workday.com`.
Tenant Name	Enter the tenant name of the account created for Oracle Integration on Workday. See Configure Workday.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Workday Username Token Policy**.
Workday Integration User	Enter the user name of the account created for Oracle Integration on Workday. See Configure Workday.
Password	Enter the password of the account created for Oracle Integration on Workday.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Configure the Oracle ServiceNow Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:
   - In the **ServiceNow Instance Name** field, enter your ServiceNow instance URL. For example, `https://instance_name.service-now.com`.
3. In the Security section, enter the following details:
   - In the **Username** and **Password** fields, enter your ServiceNow username and password.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**.

Configure the Lookup Table

The recipe contains the following two lookup tables. Edit them as necessary.

- **ORACLE-BRT-LOOPCOUNT_LOOKUP**: Use this lookup to configure the loop count, Workday EIB report name, and employee types.

Property	Description	Value
LoopCount	Specify the maximum number of iterations for a loop.	`10`
WorkdayReportName	Specify the name of the Workday EIB report.	`EIB_new_hire_report`
EmployeeType	Specify the type of the employee.	`Employee_ID`

- **ORACLE-BRT-WD_SERVICENOW_LOOKUP**: Use this lookup to map the priority and department between Workday and ServiceNow.

Property	Description	Value	WD_Value	SN_Value
Priority	Priority Values mapped between Workday and ServiceNow.	2-High	Sales	Sales

1. In the Lookups section, click the lookup name.
2. In the **Value** column, enter or map the appropriate values for the properties.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**  .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Update property values for the **Oracle Workday ServiceNow Employee Sync** integration flow. You can update the integration properties only after activating this flow.
   a. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
   b. In the Update Property Values dialog, update the following properties

Integration Property	Description
EmailTo	Specify the email address to which run-time exception emails are sent. Enter an email address of your choice.
RetryCount	Specify the number of possible retries if any record fails while creating resources at run time.
workdayTimeZoneDuration	This integration property converts Oracle Integration time zone to Workday tenant time zone. User can configure this value.

   c. Click **Submit**.

A message confirms that the integration property has been updated successfully.

3. Run the integration **Oracle Workday ServiceNow Employee Sync** to create or modify users in ServiceNow according to the employee data received from Workday. This integration triggers the **Oracle REST Workday ServiceNow** integration.

   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

      The Configure and run page is displayed, where you can specify a value for the `LastRunTime` parameter.

      The `LastRunTime` parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Workday employee records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, `yyyy-MM-dd HH:mm`. To fetch all the employee records from the beginning, set the value as `zero`.

   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

      > **Note:**
      >
      > You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

4. Run the integration **Oracle Workday ServiceNow Emp Onboarding** to create tickets in ServiceNow for the failed Workday employee records. This integration triggers the **Oracle REST Workday ServiceNow** integration.

   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the integration flow, then select **Run**.

   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

5. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.

6. Log in to ServiceNow and check for the employee records created or modified and the corresponding tickets created.

   a. View the employee records.

      i. In the **Filter** box, search for **Users**.

      ii. Under **Organization**, select **Users**.

      You can see the list of users created or modified.

   b. View the tickets created.

      i. In the **Filter** box, search for **HR Case**.

      ii. Under **All HR Cases**, click **All**.

      You can see a list of all the tickets created.

Related Documentation

- *Using the Workday Adapter with Oracle Integration 3*

- *Using the ServiceNow Adapter with Oracle Integration 3*

Sync Opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite Orders

Use this recipe to create a new sales order in Oracle NetSuite when an opportunity is won in Oracle CX Sales and B2B Service.

Topics:

About This Recipe

This recipe synchronizes opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite sales orders in real time. It uses the standard Oracle CX Sales and B2B Service Adapter and the Oracle NetSuite Adapter. To use the recipe, you must install the recipe and configure the connections and other resources within the recipe.

When an opportunity closes with the status Won in Oracle CX Sales and B2B Service, the main integration flow of the recipe is triggered. From Oracle CX Sales and B2B Service, this integration flow receives the customer and inventory item details associated with the opportunity won. Subsequently, the main flow calls two subsidiary integration flows to fetch the corresponding customer and inventory item details from Oracle NetSuite. The subsidiary flows check if the particular customer and inventory item records exist in Oracle NetSuite. If the records exist, the subsidiary flows return the internal IDs of the records to the main flow. If records don't exist (that is, if it's a new customer or a new inventory item), the subsidiary flows create a new customer record and/or a new inventory item record in Oracle NetSuite and return the internal IDs of these records to the main flow. After it receives the necessary details, the main flow creates a new sales order in Oracle NetSuite, thereby synchronizing the data between Oracle CX Sales and B2B Service and Oracle NetSuite.

> **Note:**
>
> - You can run this recipe for one Oracle CX Sales and B2B Service product group at a time.
> - For successful execution of the recipe, you must add only one product group to the opportunity in Oracle CX Sales and B2B Service, and you must not add a product item to the opportunity.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39447) or higher
- Oracle NetSuite, Version 2018.1.0
- Oracle CX Sales and B2B Service

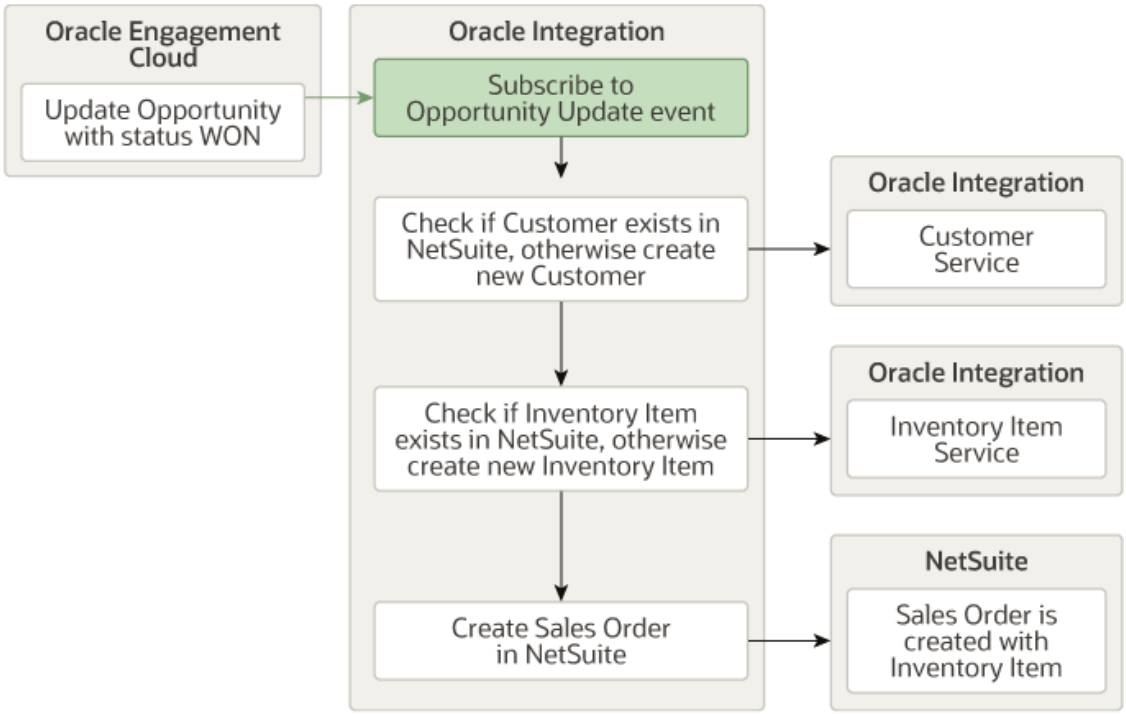
- An account on Oracle NetSuite with the Administrator role
- Accounts on Oracle CX Sales and B2B Service with the Administrator role and the Sales User role (in order to create and close opportunities)

Recipe Schemas

This section describes the integration flows present in the recipe.

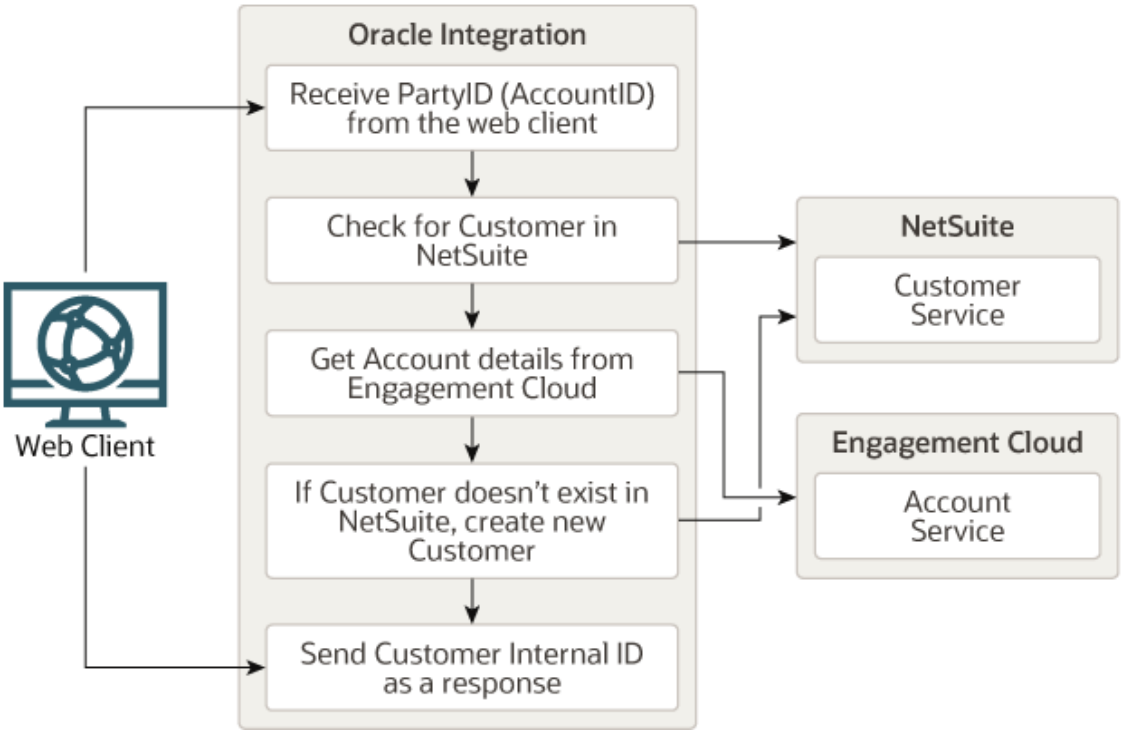
Main Integration Flow - Create Sales Order in NetSuite

This integration flow is triggered when an opportunity is closed with the status *Won* in Oracle CX Sales and B2B Service. It receives the customer and inventory item details associated with the opportunity won, and it calls the subsidiary integration flows to fetch the corresponding customer and inventory item details from Oracle NetSuite. After it receives the details, the main flow creates a new sales order in Oracle NetSuite.



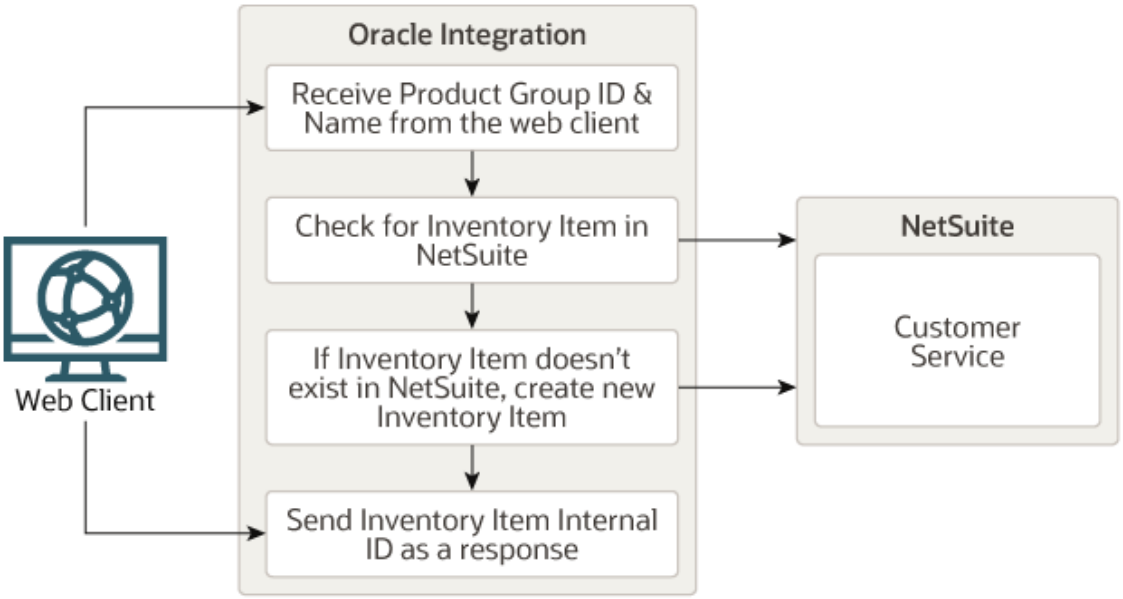
Subsidiary Integration Flow - Fetch Customer Record from NetSuite

This integration flow is triggered by a REST service from the main integration flow. As a parameter, it receives the `PartyId` (`AccountId`) of the customer record associated with the opportunity won in Oracle CX Sales and B2B Service. Using this parameter, the flow checks if the corresponding customer record exists in Oracle NetSuite. If the record exists, it returns the internal ID of the record to the main flow. If the record doesn't exist (that is, if it's a new customer), the flow fetches the account details of the customer record from Oracle CX Sales and B2B Service and creates a new customer record in Oracle NetSuite. Finally, it returns the internal ID of this record to the main flow.



Subsidiary Integration Flow - Fetch Inventory Item Record from NetSuite

This integration flow is triggered by a REST service from the main integration flow. As parameters, it receives `prodgroupid` and `prodgroupname` of the inventory item associated with the opportunity won in Oracle CX Sales and B2B Service. Using these parameters, the flow checks if the corresponding inventory item record exists in Oracle NetSuite. If the record exists, it returns the internal ID of the record to the main flow. If the record doesn't exist (that is, if it's a new inventory item), the flow creates a new inventory item record in Oracle NetSuite and returns the internal ID of this record to the main flow.



Before You Install the Recipe

You must perform the following configuration tasks on your Oracle CX Sales and B2B Service instance and Oracle NetSuite instance in order to successfully connect to these external systems using Oracle Integration and achieve synchronization.

Configure Oracle CX Sales and B2B Service

To access the Oracle CX Sales and B2B Service instance from Oracle Integration, you'll require a separate user account on Oracle CX Sales and B2B Service. In addition, you must create a Credential Store Framework (CSF) key forOracle Integration in Oracle SOA Composer to subscribe to events in Oracle CX Sales and B2B Service.

Perform the following tasks to configure Oracle CX Sales and B2B Service for this recipe.

1. Create a new user account for Oracle Integration. You'll use the credentials of this user account to connect to Oracle CX Sales and B2B Service from Oracle Integration. For the procedure to create the integration user account, see Creating an Integration User Account in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
    a. Assign the following roles to the user account. For steps to assign roles, see Assign Integration Roles in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
        - `Integration Specialist`
        - `AttachmentsUser`
        - `FND_MANAGE_CATALOG_SERVICE_PRIV`
    b. Make a note of the user name and password you set for the account.
2. Create a CSF key for Oracle Integration.
    a. Assemble the CSF Key on your Oracle Integration instance and register the same on Oracle SOA Composer. The CSF key entry in the Oracle CX Sales and B2B Service infrastructure stores the Oracle Integration credentials used by Oracle CX Sales and B2B Service. When Oracle Fusion Applications send outbound requests to Oracle Integration (at runtime), these credentials (user name and password) are sent for authentication. For the procedure to create the Oracle Integration CSF key, see Enabling Event Subscriptions in the Oracle CX Sales and B2B Service in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
    b. Create the CSF key by providing your Oracle Integration account credentials.
    c. Ensure that your Oracle Integration account is assigned at least the `ServiceInvoker` role on the Oracle Integration instance.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.

2. Perform the TBA-related configuration tasks. See Prerequisites for the Token-Based Authentication Security Policy in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.

   Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Transactions	Adjust Inventory	Full
Transactions	Cash Sale	Full
Transactions	Cash Sale Refund	Full
Transactions	Customer Deposit	Full
Transactions	Customer Payment	Full
Transactions	Customer Refund	Full
Transactions	Fulfill Orders	Full
Transactions	Invoice	Full
Transactions	Item Shipment	Full
Transactions	Sales Order	Full
Transactions	View Payment Events	Full
Reports	Integration	View
Reports	Inventory	View
Reports	Sales	View
Lists	Accounts	Full
Lists	Cases	Full
Lists	Companies	Full
Lists	Contacts	Full
Lists	Custom Record Entries	Full
Lists	Customers	Full
Lists	Employees	Full
Lists	Items	Full
Lists	Perform Search	Full
Setup	REST Web Services	Full
Setup	SOAP Web Services	Full
Setup	User Access Tokens	Full

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle CX Sales and B2B Service Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: `https://customer_chosen_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
Username	Enter the user name of the account created for Oracle Integration on the Oracle CX Sales and B2B Service. See Configure Oracle CX Sales and B2B Service .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, `https://webservices.netsuite.com/wsdl/<NetSuite_application_version>/netsuite.wsdl`.. where `<NetSuite_application_version>` is the version of the Oracle NetSuite application. For example, `v2015_1_0`. See Assemble the Oracle NetSuite WSDL URL in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Token Based Authentication**.
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Create an Integration Record for Oracle Integration in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See Create an Access Token for the User Account in *Using the Oracle NetSuite Adapter with Oracle Integration 3*.
Token Secret	Enter the token secret provided by Oracle NetSuite.

Field	Information to Enter
Account ID	Enter your Oracle NetSuite account identifier. See Make a Note of the NetSuite Account ID in *Using the Oracle NetSuite Adapter with Oracle Integration 3*. **Note:** You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**.

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back**.

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. Log in to the Oracle CX Sales and B2B Service instance using a separate account, for example, as a sales user. Click **Opportunities** on the home page.
   b. On the Opportunities page, click **Create Opportunity**.
   c. On the Create Opportunity page:
      i. Enter a name for the opportunity.
      ii. Select an account for the opportunity (mandatory).
      iii. Set a close date, and click **Save and Continue**.
   d. On the Summary page:
      i. Select **Won** in the **Status** field. Choose a suitable reason in the **Win/Loss Reason** field.
      ii. Scroll to the Products section, and click **Add** to add a product group.

iii. With **Group** selected in the **Type** field, select a product group in the **Name** field. Enter other necessary details for the product group, such as quantity, estimated price, and so on.

> **Note:**
>
> For successful execution of the recipe, you must assign only one product group to the opportunity, and you must not add a product item to the opportunity.

iv. Click **Save and Close**.

3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Log in to your Oracle NetSuite instance as an Administrator and check for the new sales order, customer, and inventory item records.
   a. To view the sales order: On the Oracle NetSuite home page, select **Transactions**, then **Sales**, then **Enter Sales Orders**, and then **List**.
   b. To view the customer record: On the Oracle NetSuite home page, select **Lists**, then **Relationships**, and then **Customers**.
   c. To view the inventory item record: On the Oracle NetSuite home page, select **Lists**, then **Accounting**, and then **Items**.

Related Documentation

- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
- *Using the Oracle NetSuite Adapter with Oracle Integration 3*

Sync Oracle Service Cloud Incidents with Oracle CPQ Transactions

Use this recipe to synchronize Oracle Service Cloud (RightNow) incidents with Oracle Configure, Price, and Quote Cloud (Oracle CPQ) transactions.

Topics:

About This Recipe

This recipe synchronizes Oracle Service Cloud incidents with Oracle CPQ transactions. To use the recipe, you must install the recipe and configure connections and other resources within the recipe. Subsequently, you can activate the integration flows of the recipe. The recipe contains three integration flows.

When an incident is created in Oracle Service Cloud, the first integration is triggered and it creates a corresponding transaction with required information such as customer details and service agent details in Oracle CPQ. This in turn triggers the second integration that updates the Oracle Service Cloud incident with the transaction Id of the corresponding transaction that

was created in Oracle CPQ. When the incident is updated with the transaction Id and the service agent changes the status of the incident to *Solved* in Oracle Service Cloud, it triggers the third integration which changes the corresponding transaction status to *Canceled* in Oracle CPQ.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Oracle Service Cloud
- Oracle CPQ
- An account on Oracle Service Cloud with the Administrator role
- An account on Oracle CPQ with the Administrator role
- Access to the Oracle Service Cloud Console

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Service Cloud and Oracle CPQ instances in order to successfully connect to these external systems using Oracle Integration and synchronize Oracle Service Cloud incidents with Oracle CPQ transactions.

Configure Oracle Service Cloud

To access Oracle Service Cloud using Oracle Integration and ensure that Oracle Service Cloud incidents and Oracle CPQ transactions are synchorinized, you must perform certain configuration tasks in your Oracle Service Cloud instance.

Log in to your Oracle Service Cloud instance as an **Administrator** and execute the following tasks.

1. Complete the prequisite tasks for creating a connection between Oracle Service Cloud and Oracle Integration. See Prerequisites for Creating a Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.
2. Add or edit an incident custom field.
    a. In the navigation pane, click **Configuration**.
    b. Expand **Database**, and double-click **Custom Fields**.
    c. To create a new incident custom field, click **Incidents** in **Custom Fields**, and then click **New**.
    d. Alternatively, to edit a specific incident custom field, click it in the tree.
    e. In the **Name** field, enter the name of the custom field. For example, `CPQTransaction`.
    f. Click **Data Type** and select the data type for the field.
3. Add the incident custom field to the Incident page.
    a. In the navigation pane, click **Configuration**.
    b. Expand **Application Appearance**, and double-click **Incident workspace**.
    c. In the Insert Field section, select the incident custom field that you created/edited in step 2, and drag it to the **Incident workspace**.
    d. Click **Save**.
4. Add fields to a custom object.

a. In the navigation pane, click **Configuration**.

b. Expand **Service**, and double-click **Object Designer**.

c. Select a package to which you want to add the custom object, or create a new package for the object by clicking **New** and selecting **Package**. Enter a name for the package.

d. Click **Fields** and then click **Add New Field** to add a customized field to the custom object.

For example, you can add the following custom fields:

Field	Type
title	text
office_phone	text
mobile	text
email	text

e. Click **Deploy**.

5. Store values to the custom fields.

a. In the navigation pane, click **Configuration**.

b. Expand **Staff Management**, select and double-click **Staff Accounts by Profile**.

c. Choose an agent from the list. Double-click the agent who will be responsible for creating incidents for customers.

d. Click **Custom Fields**, and enter values for the custom fields (title, office_phone, mobile, and email) that you created.

e. Click **Save**.

6. Add products to Oracle Service Cloud in order to ensure that those products are available in Oracle CPQ.

a. In the navigation pane, click **Configuration**.

b. Expand **Service** and double-click **Products**, then **Categories**, and then **Dispositions**.

c. To add products, click **New** and then select **Sibling or Child**.

d. Enter the name of the product. Note that the product name must be a combination of part number and name of the product. For example, `AS09743-ExampleProduct`.

e. Set Visibility to **Admins** and click **Save**.

Configure Oracle CPQ

To access Oracle CPQ using Oracle Integration you must perform certain configuration tasks in your Oracle CPQ instance.

Log in to your Oracle CPQ instance as an **Administrator** and execute the following tasks.

1. Create a user account for Oracle Integration with the **Web Services Only** permission.

a. On the Oracle CPQ home page, click **Admin Drawer** on the title bar.

b. In the Admin navigation pane, expand **Users** and click **Internal Users**.

The User Administration List page appears.

c. Add the user from the User Administration List page.

For details, see Setting Up Users.

2. Obtain the REST Catalog URL. See Prerequisites for Creating a Connection in *Using the Oracle CPQ Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See Install a Recipe or Accelerator.

After you've installed the recipe, configure the following resources:

Configure the Oracle Service Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the WSDL to use. See Configure Oracle Service Cloud.
3. In the Security section, enter the username and password that you obtained. See Configure Oracle Service Cloud.

   Note that only the Username Password Token security policy is supported.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click **Go back**  .

Configure the Oracle CPQ Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select **REST Catalog URL**. See Configure Oracle CPQ.
Connection URL	Enter the instance URL of Oracle CPQ.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **Username Password Token**.
User Name	Enter the username of the account created for Oracle Integration in Oracle CPQ. See Configure Oracle CPQ.
Password	Enter the password of the account created for Oracle Integration in Oracle CPQ.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**  .

Update Integration Property

You have to update the integration property value so that Oracle CPQ quote approval notification messages are sent to the right Slack channel.

1. In the Integrations section, click **Actions** ⋮ on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, click **pricebookname**.
3. In the **New Value** field, enter the name of the pricebook. For example, `default_price_book`.
4. Click **Submit**.

   A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See Activate a Recipe or Accelerator.
2. Run the recipe.
   a. Log in to the Oracle Service Cloud instance.
   b. As the Service agent, create an incident with customer details.

      You have now successfully triggered the recipe.
   c. Refresh the Oracle Service Cloud incident and check if it's updated with the Transaction Id.
   d. If the incident is updated with the Transaction Id, change the status of the Oracle Service Cloud incident to *Solved*. From the drop-down list of the incident's **Status** field, select **Solved**.
3. Monitor the running of the integration flow in Oracle Integration. See Monitor Integrations.
4. Check if the corresponding transaction is canceled in Oracle CPQ.
   a. Log in to the Oracle CPQ instance.
   b. Navigate to the transaction that was created corresponding to the Service Cloud incident.
   c. Check if the status of the transaction is *Canceled*.

Related Documentation

- *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
- *Using the Oracle CPQ Adapter with Oracle Integration 3*