Oracle® Cloud Getting Started with Oracle Integration Generation 2





Oracle Cloud Getting Started with Oracle Integration Generation 2,

F31221-33

Copyright © 2020, 2022, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

		•	_	_	_
ப	re	١T	ลเ	_	$^{\circ}$
		- 1 (Г.

	Audience	ix
	Documentation Accessibility	Ki
	Diversity and Inclusion	ix
	Related Resources	>
	Conventions	>
1	Welcome to Oracle Integration	
	A Quick Introduction to Oracle Integration	1-1
	Connect to Everything	1-2
	Deployment Flexibility	1-2
	Library of Prebuilt Adapters	1-3
	Integration Accelerators and Recipes	1-4
	Embedded File Server	1-4
	Business-to-Business Communications	1-6
	Automate and Optimize End-to-End Business Processes	1-7
	Example 1: Travel Approval Process	1-7
	Example 2: Accounts Payable Process	1-8
	Robotic Process Automation	1-11
	Gain Real-Time Insight into Business Processes	1-11
	Build Web and Mobile Apps in Minutes	1-12
	Steps to Build a Mobile App	1-13
	Supported Browsers	1-16
)	Navigate and Explore	
	Get Familiar with the Home Page	2-1
	View Oracle Integration Announcements	2-3
	Navigate Anywhere	2-4
	Try an Integration Recipe or a Process QuickStart	2-6
	Get Stats at a Glance	2-7
	Explore Accelerators and Recipes	2-8



Open Recently Worked on Items	2-10
See Your Process Tasks	2-10
Start Developing by Feature	2-10
Questions? Ask Oracle Assistant	2-11
Get Started with Integration Accelerators and Recipes	
Learn About Differences Between Accelerators and Recipes	3-1
Find Accelerators and Recipes	3-2
Get More Details About Accelerators or Recipes	3-2
Install Accelerators and Recipes	3-3
Configure Resources	3-3
View Details About a Resource	3-6
Activate All the Integrations in the Accelerator or Recipe	3-6
Uninstall Accelerators and Recipes	3-6
Recipes Reference	
Cloud Storage/Content Management Recipes	A-1
Export Messages from Apache Kafka to Amazon S3	A-1
Before You Install the Recipe	A-3
Install and Configure the Recipe	A-5
Activate and Run the Recipe	A-8
Customer Relationship Management (CRM) Recipes	A-9
Attach Files to Salesforce Records	A-10
Before You Install the Recipe	A-11
Install and Configure the Recipe	A-13
Activate and Run the Recipe	A-15
Create and Retrieve Organization Details from Oracle Service Cloud (RightNow)	A-16
Create Box Folders for ServiceNow Incidents	A-17
Before You Install the Recipe	A-19
Install and Configure the Recipe	A-21
Activate and Run the Recipe	A-23
Create Box Folders for Zendesk Organizations	A-25
Before You Install the Recipe	A-26
Install and Configure the Recipe	A-27
Activate and Run the Recipe	A-29
Create Orders in Oracle B2B Service for Oracle CPQ Opportunities	A-30
Create Salesforce Cases for Jira Issues	A-31
Before You Install the Recipe	A-32
Install and Configure the Recipe	A-35
Install and Configure the Recipe	A-3



Activate and Run the Recipe	A-37
Create Zendesk Tickets for Jira Issues	A-39
Before You Install the Recipe	A-40
Install and Configure the Recipe	A-41
Activate and Run the Recipe	A-43
Export Invoices from PayPal to an FTP Server	A-44
Before You Install the Recipe	A-45
Install and Configure the Recipe	A-46
Activate and Run the Recipe	A-49
Import Marketo Leads to Oracle Autonomous Data Warehouse	A-50
Before You Install the Recipe	A-50
Install and Configure the Recipe	A-51
Activate and Run the Recipe	A-54
Integrate Oracle Internet of Things Intelligent Applications Cloud with Oracle Field Service Cloud	A-55
Manage ServiceNow Incidents	A-56
Post Slack Notifications for New Marketo Leads	A-60
Before You Install the Recipe	A-61
Install and Configure the Recipe	A-64
Activate and Run the Recipe	A-66
Post Slack Notifications for ServiceNow Activities	A-67
Before You Install the Recipe	A-68
Install and Configure the Recipe	A-71
Activate and Run the Recipe	A-73
Send Compliance Documents from DocuSign to New ServiceNow Users	A-76
Before You Install the Recipe	A-77
Install and Configure the Recipe	A-80
Activate and Run the Recipe	A-82
Send Documents from Salesforce to DocuSign	A-84
Before You Install the Recipe	A-85
Install and Configure the Recipe	A-90
Activate and Run the Recipe	A-92
Send Notifications to Salesforce Lead Owners Using Twilio	A-93
Before You Install the Recipe	A-94
Install and Configure the Recipe	A-98
Activate and Run the Recipe	A-100
Send SurveyMonkey Surveys to Salesforce Contacts by Email	A-102
Before You Install the Recipe	A-102
Install and Configure the Recipe	A-108
Activate and Run the Recipe	A-110
Sync Customer Data Between Oracle Loyalty Cloud and Oracle Responsys	A-112



	Sync Data Between Oracle CPQ Transactions and Salesforce Quotes	A-113
	Before You Install the Recipe	A-114
	Install and Configure the Recipe	A-123
	Activate and Run the Recipe	A-127
	Sync Incidents and Resources Between Oracle Field Service Cloud and Oracle B2C Service	A-129
	Sync Orgs and Contacts Between Oracle B2C Service and Oracle Marketing Cloud	A-130
En	nterprise Messaging Recipes	A-131
	Publish and Subscribe to the Oracle Cloud Infrastructure Streaming Service	A-131
En	nterprise Resource Planning (ERP) Recipes	A-134
	Automate File Transfer an FTP server to Oracle EPM Cloud	A-135
	Create Customer Records in Oracle NetSuite	A-136
	Before You Install the Recipe	A-136
	Install and Configure the Recipe	A-137
	Activate and Run the Recipe	A-139
	Export Purchase Orders from Oracle ERP Cloud to an FTP Server	A-140
	Extract Bulk Data from Oracle ERP Cloud to an FTP server	A-144
	Import Files from an FTP Server to Oracle NetSuite	A-145
	Before You Install the Recipe	A-146
	Install and Configure the Recipe	A-147
	Activate and Run the Recipe	A-150
	Import Financial Journal Entries from Amazon S3 to Oracle ERP Cloud	A-151
	Import Financial Journal Entries from an FTP server to Oracle ERP Cloud	A-152
	Process Inventory, Order, and Shipping Info Between Oracle ERP/SCM Cloud and Oracle WMS Cloud	A-153
	Send Order Fulfillment Notifications from Oracle ERP Cloud Using Microsoft Office 365 Outlook	A-154
	Before You Install the Recipe	A-155
	Install and Configure the Recipe	A-157
	Activate and Run the Recipe	A-159
	Synchronize Inventory Items Between Oracle Product Hub Cloud and Oracle E- Business Suite	A-160
	Before You Install the Recipe	A-162
	Install and Configure the Recipe	A-163
	Activate and Run the Recipe	A-165
	Synchronize Opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite Orders	A-167
	Before You Install the Recipe	A-170
	Install and Configure the Recipe	A-172
	Activate and Run the Recipe	A-174
	Synchronize Supplier Updates Between Oracle Procurement Cloud and Oracle ATP	A-176
	Before You Install the Recipe	A-177



Install and Configure the Recipe	A-181
Activate and Run the Recipe	A-183
Human Capital Management (HCM) Recipes	A-184
Export Employee Data from Oracle HCM Cloud to an Identity Management System	A-184
Import Users from Oracle HCM Cloud to Okta	A-185
Import Worker Records from an FTP Server to Oracle HCM Cloud	A-186
Transform an Oracle HCM Cloud Extract File to an ANSI 834 Benefits File	A-187
Industries Recipes	A-188
Generate Oracle Hospitality Access Tokens	A-188
Install and Configure the Recipe	A-189
Activate and Run the Recipe	A-190
Oracle Integration Starter Recipes	A-191
Calculate Total Balance	A-192
Concatenate Messages	A-194
Echo Request	A-195
Get Incident Details	A-196
Get Weather Data	A-196
Increment the Sequence Variable within a Loop Integration	A-198
Perform Mathematical Operations	A-200
Get Stocks Data	A-210
Say Hello World	A-212
Say Hello World Invoke	A-213
Say Hello World Map Data Invoke	A-213
Send Files from Your Application by Email	A-214
Install and Configure the Recipe	A-215
Activate and Run the Recipe	A-215
Transfer a File	A-216
Use Multiple Verbs and Resources Invoke	A-218
Work with Files on FTP Servers	A-219
Social and Productivity Recipes	A-220
Export Google Gmail Attachments to an FTP Server	A-220
Before You Install the Recipe	A-221
Install and Configure the Recipe	A-222
Activate and Run the Recipe	A-224
Integrate with DocuSign	A-225
Interact with Facebook	A-228
Interact with LinkedIn	A-232
Interact with Slack	A-234
Interact with Twitter	A-238
Post Slack Notifications for Completed SurveyMonkey Surveys	A-241
Before You Install the Recipe	A-241



Install and Configure the Recipe	A-245
Activate and Run the Recipe	A-247
Synchronize SurveyMonkey with Oracle NetSuite	A-248
Other Recipes	A-249
Technical Accelerators Reference	
Alert Notification	B-1
Description	B-1
Key Parameters	B-1
Supported Notification Types	B-2
Connections	B-2
Lookups	B-2
Usage	B-3
Resequence Messages	B-4
Description	B-4
Prerequisite	B-5
Key Parameters	B-5
Connections	B-6
Architecture	B-6
Error Handling	B-9
	Activate and Run the Recipe Synchronize SurveyMonkey with Oracle NetSuite Other Recipes Technical Accelerators Reference Alert Notification Description Key Parameters Supported Notification Types Connections Lookups Usage Resequence Messages Description Prerequisite Key Parameters Connections Architecture



Preface

This document describes how business analysts and integration specialists get started working with Oracle Integration Generation 2.

Topics:

- Audience
- Documentation Accessibility
- Related Resources
- Conventions

Audience

This guide is intended for business analysts and developers who want to get an overview of Oracle Integration Generation 2, what it can do, and how to use recipes to build their first 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 customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup? ctx=acc&id=info Or Visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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 in the Oracle Cloud Library 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 fo 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

Welcome to Oracle Integration

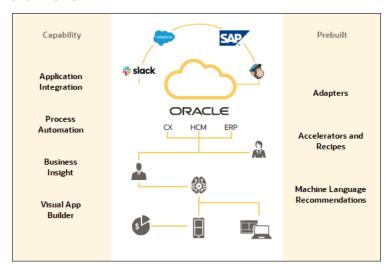
Oracle Integration is a secure, unified platform that lets you connect cloud and on-premises applications, automate business processes, gain insight into your business through business metrics analysis, and develop web and mobile applications.

Topics:

- A Quick Introduction to Oracle Integration
- Connect to Everything
- Automate and Optimize End-to-End Business Processes
- Gain Real-Time Insight into Business Processes
- Build Web and Mobile Apps in Minutes
- Supported Browsers

A Quick Introduction to Oracle Integration

Integration is a fundamental part of your digital business development. It involves connecting on-premises applications and cloud applications and services. Oracle Integration makes it easy to connect your applications and automate end-to-end processes such as procure to pay, inquiry to order, and hire to retire.



With Oracle Integration, you can:

- Develop integrations to design, monitor, and manage connections between your applications
- Create process applications to automate and manage your business work flows
- Analyze results to gain insight into your business
- Build custom web and mobile applications

Critical business processes, such as those related to human capital management (HCM), customer experience (CX), and enterprise resource planning (ERP), are frequently slow and inflexible. For example, a multi-step process such as Lead to Opportunity to Quote to Order can involve four or more applications and require human exception management at every step of the process. In this scenario, the lack of integration between departments as well as the delays caused by human-based problem resolution can result in lost revenue, frustrated customers, and high costs.

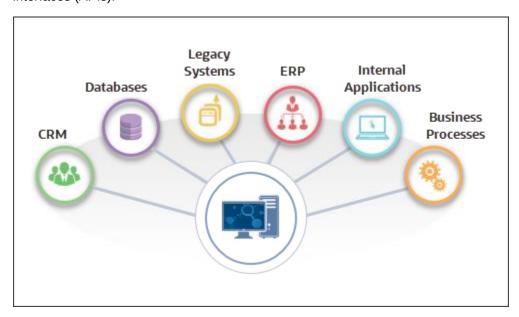
Oracle Integration changes all that. It empowers you to:

- Establish connectivity between the many applications and people that are part of the entire business process life cycle.
- Assemble existing technologies into new business services to better align with the changing pace of new business demands.
- Deliver new business innovations faster by rapidly connecting diverse applications and key business roles.
- Gain 360-degree views across your entire business. Easily monitor and analyze every application, integration, and workflow spanning the business process life cycle.

Connect to Everything

Integration solves the application silo problem. It brings together data and workflows between disparate software applications for them to function as one.

Integrations deal with live operational data. Data can be processed either in real-time or in batch, and between two or more applications, events, or application programming interfaces (APIs).



Deployment Flexibility

With integrations, you can connect to anything without getting bogged down in technical implementation details.



You can bring data and applications together across on-premise and cloud systems: cloud to cloud, cloud to on-premises (ground), and on-premises to on-premises (ground to ground).

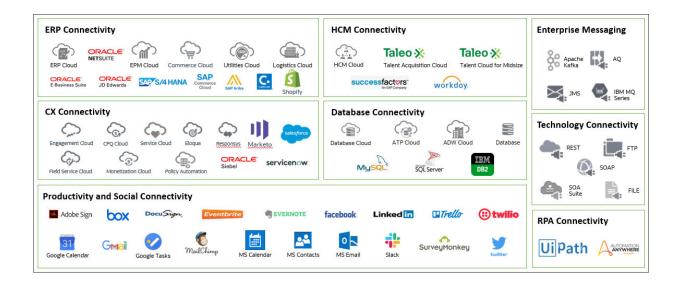


Library of Prebuilt Adapters

When you develop an integration from scratch, your first task is to create a connection for each application that you're going to connect to. The connection defines how you'll connect to the application. When you create a connection, your first step is to select an adapter.

An **adapter** provides the means for Oracle Integration to connect to different applications. If you have traveled internationally, then you have probably used a power adapter that allowed your electronic devices to connect safely with a power outlet of a different shape and voltage than what it was originally built for. Well, Oracle Integration adapters do the same thing for software.

Oracle Integration includes dozens of prebuilt adapters that make it easy for your integrations to connect to a range of Oracle and third-party applications, including applications that handle your customer relationships, human resources, supply chain, resource planning, ecommerce, social media, and databases. Also, the FTP, REST, and SOAP Adapters let you use standard file, web, and messaging protocols. And all adapters let you connect to applications that are on premises or in the cloud.



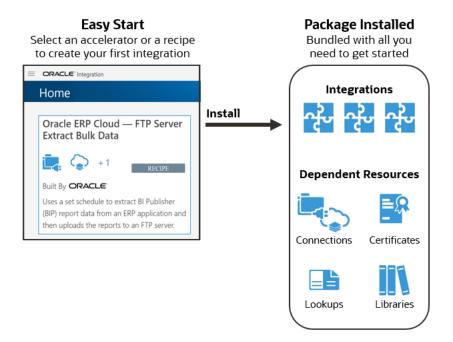
After you choose an adapter, you provide the information that lets the adapter successfully find and connect to a particular instance of an application. The information might include the application URL, the security policy to use, and the credentials for signing in to the application.



Integration Accelerators and Recipes

You don't need to start an integration project from scratch. You can install ready-made integration accelerators and recipes from the Home page. **Accelerators** are run-ready business integrations or technical patterns you can configure and activate. **Recipes** are starter templates that give you a head start.

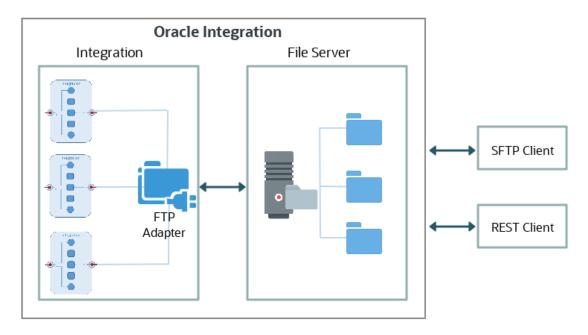
When you install an accelerator or a recipe, you get a package of everything you need to easily configure and activate your first integration. The installed package includes one or more integrations, along with all the resources required to run the integration. The dependent resources include the connections as well as any certificates, lookups, and libraries.



Embedded File Server

Use **File Server** to store files or transfer real-time data among applications securely. File Server provides an embedded Secure File Transfer Protocol (SFTP) server within Oracle Integration, so you can focus on building integrations without needing to host and maintain a separate SFTP server.





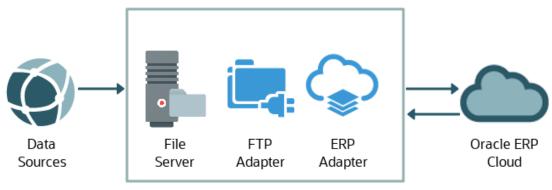
With File Server, you can:

- Design integrations that process your files that reside in the embedded file server.
- Eliminate the cost and operational expenses associated with hosting and maintaining an SFTP server by moving your SFTP server to the cloud. If you have an on-premises SFTP server, then you can move your SFTP files into File Server in Oracle Integration and use the SFTP adapter to connect.
- Give your vendors or partners access to Oracle Integration to upload and download files
 with their SFTP client software. A common protocol for communication with partners is
 SFTP. File Server enables partners to send information such as purchase orders,
 invoices, shipping information, and so on through SFTP.

Example: Read, Transform, Write

For example, if you have applications that export bulk data to an SFTP server, Oracle Integration can pick up the file, translate it into the required format, and send it to the target system. For instance, E-Business Suite generates a zip file with external transactions and uploads it to File Server. An integration can then read the file, transform it into the format required by the ERP system, and trigger bulk import of the data.

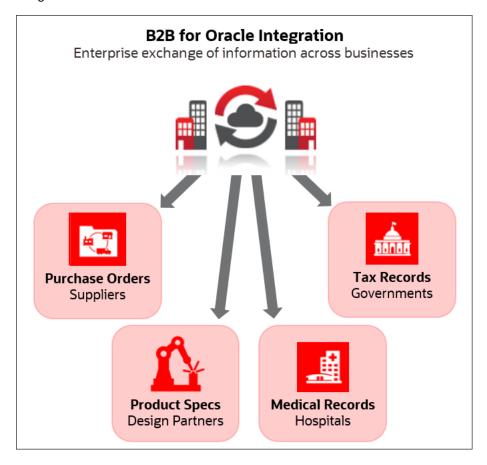
Oracle Integration





Business-to-Business Communications

B2B for Oracle Integration provides support for business-to-business (B2B) e-commerce. You can communicate with trading partners, and send and receive data in integrations with B2B.



B2B for Oracle Integration lets you:

- Securely exchange business documents, such as purchase orders or product specifications, with your trading partners using standard Electronic Data Interchange (EDI) formats.
- Securely exchange outside the enterprise with your trading partners.

In B2B e-commerce, an enterprise extends its business processes to reach trading partners, for example, suppliers, manufacturers, hospitals, and government agencies. B2B e-commerce represents classic business processes, mature business documents, and industry-tempered messaging services. It requires a unified business process platform, end-to-end instance tracking, visibility and auditing, integrated process intelligence, process and service governance, and centralized security.



Automate and Optimize End-to-End Business Processes

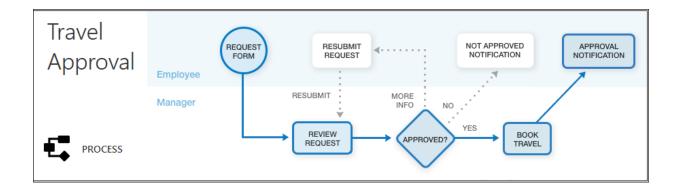
Automate and optimize any **process** in your organization. For example, you can automate core business processes such as loan origination if you're a bank, shipping and handling if you're a delivery service, or opportunity to order if you're in retail.

Topics:

- Example 1: Travel Approval Process
- Example 2: Accounts Payable Process
- Robotic Process Automation

Example 1: Travel Approval Process

Let's look at a sample process that employees use to request corporate travel.



An employee completes a travel request form and submits it to the manager. The manager reviews the travel request and takes the appropriate action. Does the manager approve the travel? The automated process handles three cases:

- **Yes**. The manager approves the travel. The process automatically sends an email that instructs the employee to book the travel.
- No. The manager doesn't approve the travel. The process automatically sends an email
 to the employee and explains why the travel was denied.
- More Info Required. The manager needs more information before making a decision.
 The travel request is returned to the employee, who must provide details about the travel and then resubmit the request. The manager reviews the travel request again and then takes the appropriate action.

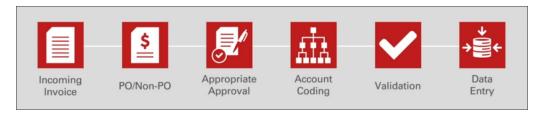


Example 2: Accounts Payable Process

Let's look at a second use case that automates the process used by the Accounts Payable department to handle incoming invoices. The goals are to eliminate lost invoices and reduce the chance for human errors.

Before Automation

Accounts Payable has to deal with a large volume of paper invoices, which can be confusing and time consuming.



When an invoice is received, the accounts payable clerk has to figure out which action to take depending on the type of invoice (purchase order or not).

The accounts payable clerk then has to send emails to the appropriate colleagues for approvals, handle account coding, validate that all required information has been completed, and at the end, manually enter the data into Oracle E-Business Suite.

After Automation

With Oracle Integration, you can streamline and automate the accounts payable process, eliminating lost invoices, reducing the time to process an invoice, and reducing the chance for human error. Through Integration Insight in Oracle Integration, management also gains visibility into the state of Accounts Payable.

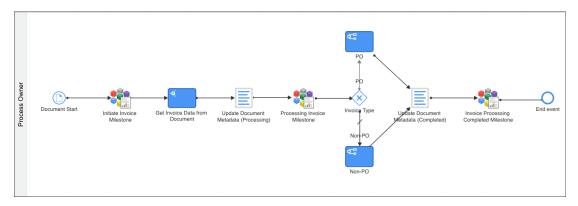
Here's the streamlined process in Oracle Integration:

- Scan the invoices or capture PDFs from emails to extract data using Optical Character Recognition (OCR) with Oracle WebCenter Forms Recognition. The extracted data is stored in Oracle Content Management along with the original PDF invoices.
- 2. From Oracle Content Management, the accounts payable clerk triggers the invoice approval workflow in Oracle Integration.
- 3. The defined approval process validates the invoice, segregates on invoice type (PO or Non-PO), and processes the invoice according to its type. A purchase order invoice is pre-approved whereas a non-purchase order invoice requires approvals.

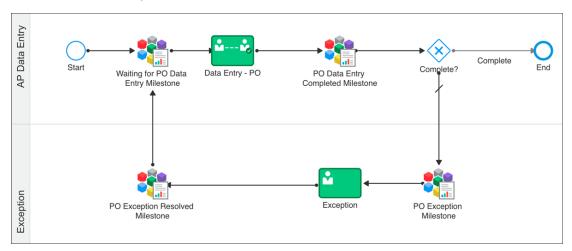
The workflow in Process in Oracle Integration has a main process and two subprocesses. One subprocess handles an invoice that's a purchase order; the other handles an invoice that's not a purchase order. Also, notice that there are defined milestones in the processes. These milestones are for analysis with Integration Insight.

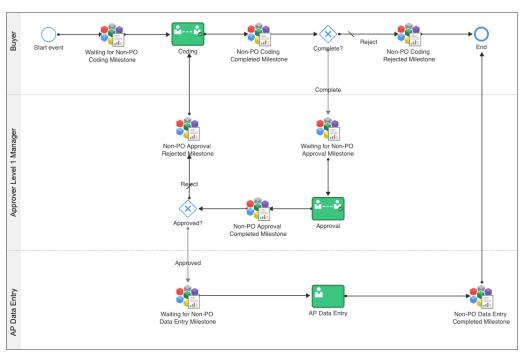


Main Process for Accounts Payable Automation



Purchase Order Subprocess

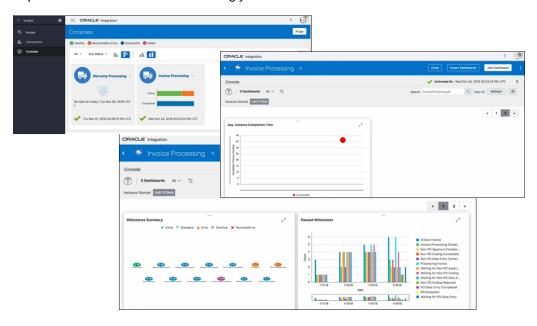




Non-Purchase Order Subprocess

Get Real-time Visibility with Integration Insight

When the Accounts Payable Director wants to know the state of the department, Integration Insight provides visibility into invoices that have passed milestones, how many invoices were rejected, how many invoices were in error, and what's the average time to process each invoice. The Director can now know the current state of the department and take action accordingly.





Robotic Process Automation

Robotic process automation (RPA) extends integration and automation to application user interfaces, bridging some of the gaps in connecting applications as well as in automating human actions. People, robots, and systems are connected end-to-end. Less manual intervention saves time, reduces errors, and frees up humans to perform more value-added jobs.

Oracle Integration combines with RPA for:

- Legacy systems that only offer connectivity through the user interface, whether it's a terminal, character-based interface, web application, or Windows application.
- Systems that don't have an API or for which there are no Oracle Integration adapters.
- On-premises and SaaS applications that have been heavily customized in the user interface but have not replicated that extension at the API level.
- Rapid automation and integration prototyping. Some organizations want to prototype automation and integration through a robot before they create a new API or service.

Example: Opportunity to Order with Robotic Data Entry

Let's suppose you want to streamline an opportunity-to-order process. For any activity within the process that doesn't require decision-making, such as data entry, give the work to a robot. Not only is the robot able to do the work automatically, but you don't need to train your sales employees to sign in to a legacy system and transact against an old system.

So, how do you handle this case with RPA? In Oracle Integration, create a process that evaluates an order with a discount. The process automatically routes the approvals to the correct people. The sales manager needs to determine whether the order can be approved with the requested discount. It's almost the end of the quarter, so the VP also needs to approve the discount. After all approvals are in place, the process triggers a robot using RPA to enter the transaction in the backend order management system.

Gain Real-Time Insight into Business Processes

Integration Insight provides real-time business visibility and analytics by allowing business milestones and metrics to be defined, aggregated, and visualized.

By defining an Integration Insight model, you can map milestones to key points in a business process implementation in integration flows or process applications (or both) in Oracle Integration, and extract metrics to allow business executives to view and analyze business processes in real time using dashboards.

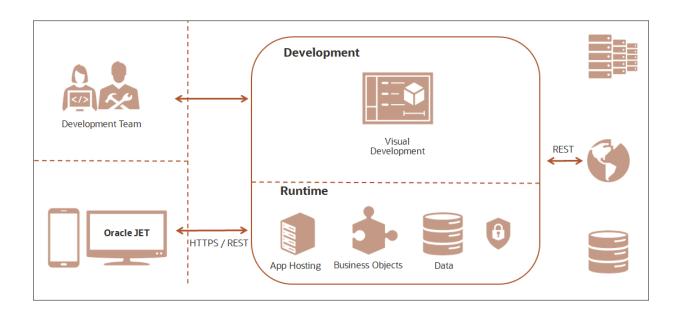
With real-time data throughout the business process lifecycle, you get a view of the business that allows you to react quickly to an issue. You have real-time visibility into digital processes for business owners to innovate smarter, and to highlight stalled transactions and errors.





Build Web and Mobile Apps in Minutes

With **Visual Builder**, you can extend your applications to meet your unique needs and build great web and mobile applications yourself. There's no coding, no setup, and no IT resources required.



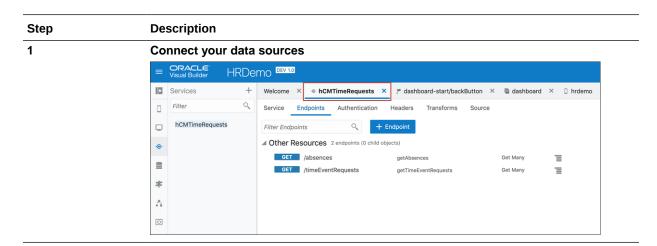
Visual Builder provides all the necessary tools for you to build, publish, and host modern web and mobile applications:

- Configure and customize Oracle Cloud applications using the same development environment that Oracle Cloud applications are built on.
- Use cloud-based visual tools to rapidly create and host web and mobile applications with minimal coding required.
 - Use the what-you-see-is-what-you-get (WYSIWYG) page designer to drag and drop UI components and visually create your pages.
 - Create custom reusable business objects that store data, implement business logic, and connect to processes.
 - Publish your application with the push of a button and make it available to users.
- Easily connect REST APIs to integrate data from other applications into yours. For more complex needs, developers can extend the functionality of the application using standard JavaScript, HTML, and CSS.

Visual Builder also provides the infrastructure for securing access to your application, data, and the Oracle Cloud services that your application consumes.

Steps to Build a Mobile App

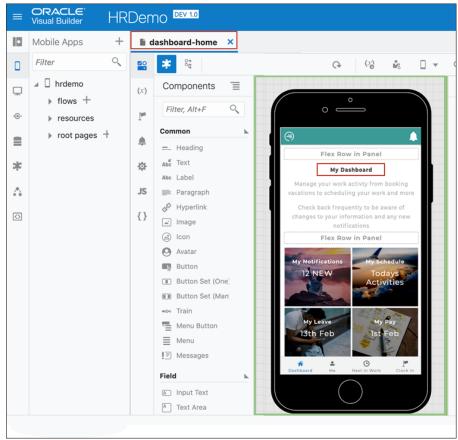
It's as easy as 1, 2, 3, ...4.





Step Description

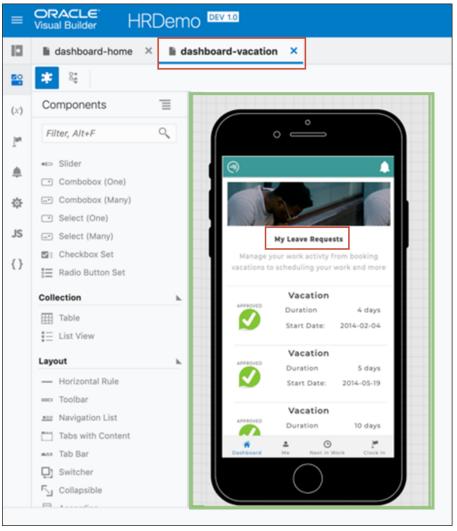
2 Create your dashboard page





Step Description

3 Display backend data



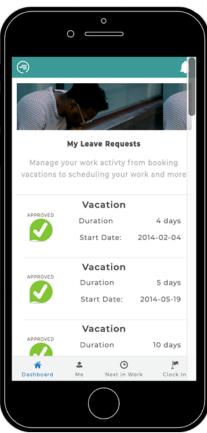


Step Description

4 Go live

And just like that, you're done. Your first mobile app built specifically to meet your needs. You can use the app on both Android and Apple iOS devices. You can share just within your company or via Google Play and Apple App Stores if needed.





Supported Browsers

Oracle Integration supports the following browsers.

For The	See
Oracle Cloud Infrastructure Console (to perform administrative tasks such as provisioning and administering your instance)	Supported Browsers
Oracle Integration user interface (to design, deploy, and monitor integrations; automate business processes; provide real-time business visibility and analytics; build web and mobile applications; and more)	What platforms are supported by Oracle JET Note: Oracle Integration supports JET version 10.



2

Navigate and Explore

After your system is provisioned and user roles are assigned, begin exploring the rich features of Oracle Integration.

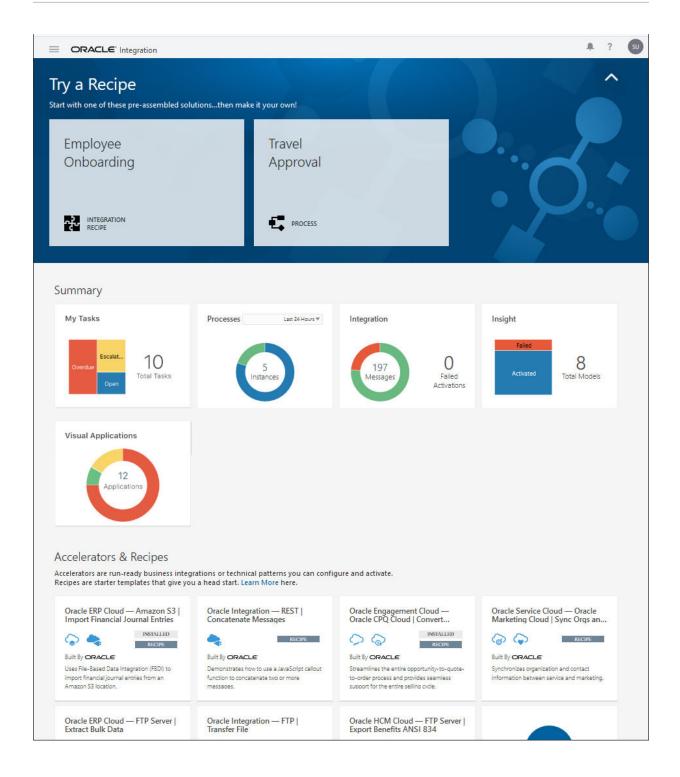
Topics:

- Get Familiar with the Home Page
- View Oracle Integration Announcements
- Navigate Anywhere
- Try an Integration Recipe or a Process QuickStart
- Get Stats at a Glance
- Explore Accelerators and Recipes
- Open Recently Worked on Items
- See Your Process Tasks
- Start Developing by Feature
- · Questions? Ask Oracle Assistant

Get Familiar with the Home Page

When you sign in to Oracle Integration, the Home page opens. From here, you can navigate to the product features you want, view a mini dashboard of relevant metrics and status, and link directly to your current tasks, applications, and work in progress.





Take a moment to get familiar with all you can do on the Home page.

Home Page Element	Description
Try a Recipe	Showcases an integration recipe for employee onboarding and a QuickStart App for a travel approval process. These two prebuilt solutions are from the extensive collection of recipes, accelerators, QuickStart Apps, and samples included in Oracle Integration. You can use any available prebuilt solution to get a head start in building integrations or process applications instead of creating them from scratch. Tip: To hide the banner and free up space on the Home page, click the Up Arrow icon.
Summary	Gives you a snapshot of key metrics for your tasks, processes,
Summary	integrations, visual applications, and Insight models. For a break down of the total numbers, position the cursor over a color. For quick access to more details, click an individual card.
Accelerators & Recipes	Lists some of the accelerators and recipes available in Oracle Integration. Accelerators and recipes jump-start your integration development and provide end-to-end connections for critical business problems. To browse the entire collection, click Search All .
Recent	Provides direct access to the integrations and process applications that you worked on recently. It's a quick and easy way to return to your work.
My Tasks	Displays the most recent process tasks assigned to you.
Go to	Jumps directly to the appropriate page and tools for developing processes, integrations, B2B integrations, visual applications, or Insight models.
New Request	Lists the process applications available to you. You can run the application and submit a request without having to navigate to the My Apps page.
Learn	Provides access to more information about Oracle Integration, including what's new, videos, and online help.
f	You can also click the Help icon in the toolbar to access videos, tutorials, and online help at any time.
ı.	Displays announcements about important events or issues concerning this Oracle Integration instance.
	Shows or hides the navigation pane and menu.
_	The Home page gives you quick access to what you do in Oracle Integration. However, it's not your only option. At any time, you can click the Navigation Menu icon in the top-left corner of the Home page to access all pages, tools, dashboards, and settings.

View Oracle Integration Announcements

View alerts about important events or issues concerning your Oracle Integration instances.

1. In the upper corner of the Home page, click — to display the Announcements window, which lists important items for your Oracle Integration instance.

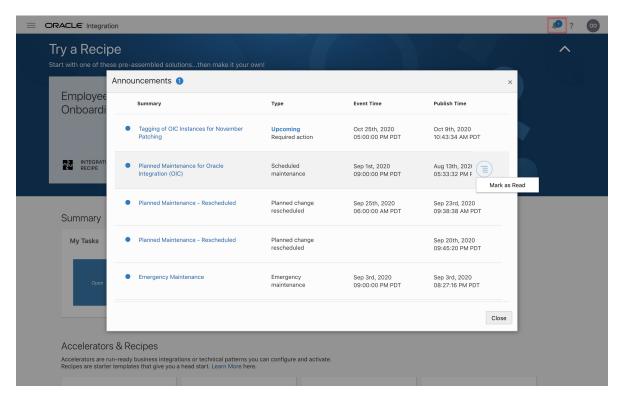
A number on the icon indicates the number of upcoming announcements that are unread. (Unread announcements that have passed their dates are not reflected in the number.)



Note:

Announcements must be enabled for your instance. If you see a message that there are no announcements, contact your Oracle Integration administrator.

In the Announcements window, items are listed by publish date, with most recent items at the top. Announcements typically indicate required actions (such as tagging) or upcoming maintenance, and the list is refreshed every hour. A blue dot appears next to unread announcements.



Mark an item as read by hovering over the item's row and selecting Mark as Read from the options menu that displays.

Navigate Anywhere

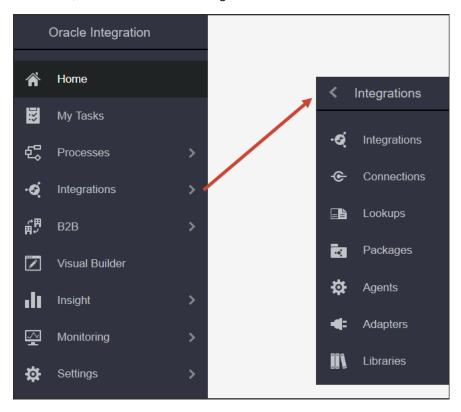
Use the navigation pane to get access to all the pages, tools, dashboards, and settings in Oracle Integration.

When you start Oracle Integration, the navigation pane is hidden. To view the navigation pane, click the **Navigation Menu** icon in the top-left corner of the Home page.





The menu displayed in the navigation pane depends on your assigned role, your selection, and your location in Oracle Integration. The navigation pane may display the main menu or a sub-menu, such as the one for Integrations.



Take a moment to learn about the main menu. It gives you access to the Oracle Integration runtime environment, design-time environment, development tools, and administration settings.

Click	То
Access the launch pad and high-level dashboard for Oracle Integration, for accelerators and recipes, and open recently worked on projects.	
My Tasks	Launch the runtime environment for process tasks. Start process applications, work on tasks as a user, view dashboards, and perform runtime administration such as mapping process roles. See Quick Tour of the Tasks Page in <i>Using Processes in Oracle Integration</i> .



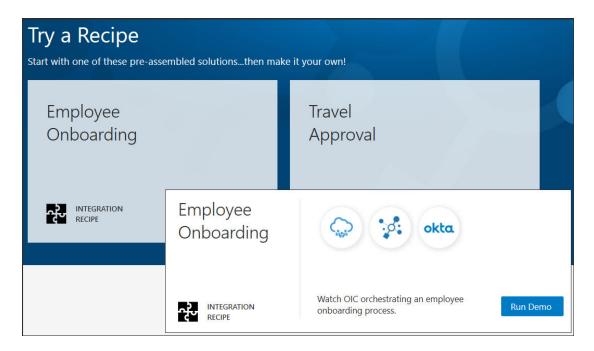
Click	То
Processes	Launch the design-time environment for process applications. Create process applications from scratch or by using QuickStart Apps and samples. Model structured and dynamic processes in the process editor, and create web forms and decision models. Perform design-time administration such as playing, testing, and deploying applications.
	See Quick Tour of the Processes Page in <i>Using Processes in Oracle Integration</i> .
Integrations	Launch the design-time environment for integrations. Configure connections, and create and activate integrations. Work with lookups, packages, agents, and adapters.
	See Getting Started with Integrations in <i>Using Integrations in Oracle Integration</i> .
用 用 B2B	Exchange business documents between Oracle Integration and a trading partner securely and reliably.
	See Introduction to B2B for Oracle Integration in <i>Using B2B for Oracle Integration</i> .
Visual Builder	Create and publish web and mobile applications.
Visual Builder	See Getting Started with Oracle Visual Builder in OIC in <i>Developing Applications with Oracle Visual Builder in Oracle Integration</i> .
Insight	Model business processes and monitor business metrics in real-time using interactive dashboards to take timely actions.
	See Introduction to Integration Insight in <i>Using Integration Insight in Oracle Integration Generation 2</i> .
Monitoring	Monitor integrations in the runtime environment. View the dashboard to see the performance of your integrations, track business identifiers associated with your integrations, monitor scheduled integration runs, or view and manage errors.
	See Viewing the Dashboard in Using Integrations in Oracle Integration.
Settings	Configure settings for error logs and manage security certificates. Specify other settings such as notification services and mapping recommendations.
-	See Configure Settings for Error Logs in Administering Oracle Integration Classic and Upload an SSL Certificate in Using Integrations in Oracle Integration.

Try an Integration Recipe or a Process QuickStart

The **Try a Recipe** banner at the top of the Home page showcases an integration recipe for employee onboarding and a QuickStart App for a travel approval process.

These two prebuilt solutions are from the extensive collection of recipes, accelerators, QuickStart Apps, and samples included in Oracle Integration. You can use any available prebuilt solution to get a head start in building integrations or process applications instead of creating them from scratch.





Integration Recipe - Employee Onboarding

To view the integration flow, position your cursor on **Employee Onboarding** and click **Run Demo**. After the demo, click **Get Started** to go to the Accelerators and Recipes page. You can now install the recipe, configure the connections, activate the integration, and explore how the integration works.

See Get Started with Integration Accelerators and Recipes.

Process QuickStart - Travel Approval

To view the process flow, position your cursor on **Travel Approval** and click **Run Demo**. After the demo, click **Get Started** to go to the Gallery page. Search for and open the Travel Approval application to customize the process for your organization.

See Create Your First Application (a Quick Start) in Using Processes in Oracle Integration.

Tips

- To hide the banner and free up space on the Home page, click the **Up Arrow** icon.
- To find other recipes that you can use to create an integration, scroll to the Accelerators
 & Recipes section on the Home page and click Search All.
- To find other QuickStart Apps that you can use to create process applications, open the
 Oracle Integration navigation pane, click Processes, and then click Process
 Applications. Click Create and then click Start with a QuickStart. The Gallery lists both
 QuickStart Apps and Sample applications that you can use as is or adapt to fit your
 business needs.

Get Stats at a Glance

The **Summary** section on the Home page, which is a mini dashboard, gives you a snapshot of key metrics for your tasks, processes, integrations, and Insight models.



To see a break down of the total numbers, position the cursor over a color. To get quick access to more details, click an individual card.



Summary Item	Description	Click to
My Tasks	Shows the total number of tasks assigned to you.	Open the My Tasks page and work on your assigned tasks.
Processes	Shows the total number of process instances that have been activated over a specified time. Use the drop-down menu to select a period of the last 24 hours, 48 hours, or 7 days.	Open the Process Applications page. On this page, you can create and activate process applications, or work on existing ones. See Tour the Process Applications Page in <i>Using Processes in Oracle Integration</i> .
	To see the number of completed processes, position the cursor over the green area.	
	To see the number of processes that are in progress, position the cursor over the blue area.	
Integrations	Shows the total number of messages and failed activations.	Open the Dashboards page and get a comprehensive view of how your integrations are performing. See View the Dashboards in <i>Using Integrations in Oracle Integration</i> .
	To see the number of failed messages, position the cursor over the red area.	
	To see the number of successful messages of active integrations, position the cursor over the green area.	
Insight	Shows the total number of Insight models.	provides access to all the models in all lifecycle states. On this page, you can create new models and work with existing models.
	To see the number of failed models, position the cursor over the red area.	
	To see the number of deactivated models, position the cursor over the yellow area.	
	To see the number of activated models, position the cursor over the blue area.	
Visual Applications	Shows the total number of visual applications. Hover over each pie section to see the number of applications in that category.	Opens the Visual Builder page, which provides access to all visual applications. On this page, you can create new applications and work with existing ones.
		See Getting Started with Oracle Visual Builder in OIC in Developing Applications with Oracle Visual Builder in Oracle Integration.

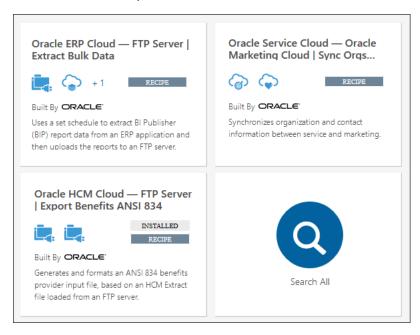
Explore Accelerators and Recipes

Oracle Integration offers a rich set of run-ready business and technical integrations called **accelerators**, and also sample templates called **recipes**. Accelerators and

recipes give you a head start in creating your integrations and provide end-to-end connections for critical business problems.

In Oracle Integration, you can group one or more integrations into a single structure called a **package**. Because you created the package, its type is **developed**. Accelerators and recipes are other types of packages. An accelerator package or a recipe package consists of a series of prebuilt integrations developed by Oracle.

The **Accelerators & Recipes** section on the Home page displays some of the available accelerators and recipes.



To learn about a particular accelerator or recipe, review the information on its card. A card:

- Lists the two applications that the integration connects to, followed by a brief purpose of the accelerator or recipe.
- Uses images and +number to indicate the adapters used by the integration. Position the cursor over the images to see the name of an adapter.
- Displays INSTALLED to indicate that the accelerator or recipe, as well as all its dependent resources, are already deployed in Oracle Integration.
- Displays ACCELERATOR or RECIPE to indicate the type of package.
- Lists who produced the accelerator or recipe.
- Provides a full description of the accelerator or recipe.

To browse the entire collection, click **Search All**. You can then search, filter, and sort the list to find the accelerator or recipe you want to use.

To get quick access to one of the accelerators or recipes listed on the Home page, click its card. Note that the Accelerators and Recipes page now only displays the accelerator or recipe you selected because the search parameters were automatically set. Clear the search criteria to see all the accelerators and recipes.

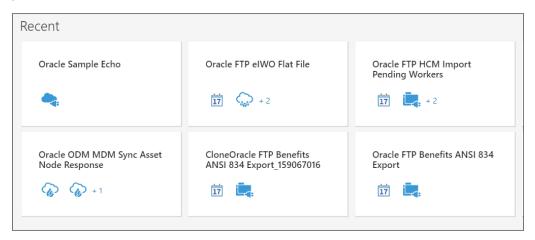
You can install the accelerator or recipe, configure its connections, and activate its integrations. See Get Started with Integration Accelerators and Recipes.



Open Recently Worked on Items

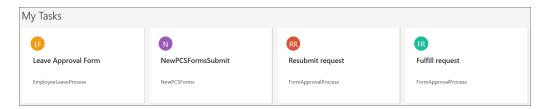
The **Recent** section on the Home page lists the items that you worked on recently. Perhaps you edited several integrations or built some new processes.

No need to navigate the menus and search for where you left off. Instead, click the card in the Recent section to return to your work. It's fast, direct, and personalized for you.



See Your Process Tasks

The My Tasks section on the Home page displays your process tasks.

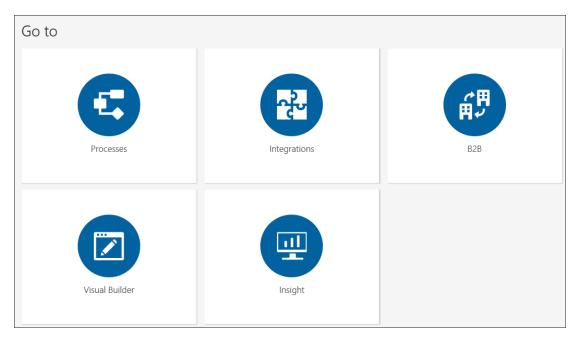


Click the task to open it in workspace, view details, or work on the task if it's assigned to you.

Start Developing by Feature

Use the **Go to** section on the Home page to jump directly to the appropriate page and tools for developing processes, integrations, B2B integrations, visual apps, or Insight models.





You can use these shortcuts instead of traversing the menu in the navigation pane. For direct access:

- Click **Processes** to create an application from scratch, or select a sample or QuickStart App from the gallery that you can customize or activate as is.
- Click **Integrations** to create connections and an integration that uses them, or activate a ready-to-use sample integration.
- Click B2B to use an Electronic Data Interchange (EDI) format in an orchestrated integration to exchange documents in a secure and reliable manner.
- Click Visual Builder to create and publish custom web and mobile applications.
- Click Insight to create and edit models that describe the milestones of a business process that are key for tracking business performance.

Questions? Ask Oracle Assistant

Oracle Assistant is a digital assistant that can answer common questions about Oracle Integration. If you have questions about Oracle Integration, ask Oracle Assistant.

Oracle Assistant joined Oracle Integration in August 2021 and was developed using Oracle Digital Assistant. You can ask questions in full sentences and Oracle Assistant will try getting the best answer for you, even searching the product documentation. You can ask general questions about Oracle Integration or specific questions about Integrations, Process, Integration Insight, File Server, B2B, or Visual Builder.

When you ask questions, try to be as specific as you can in what you're looking for. For example, if you're looking for information on three-legged Oauth configuration, tell the assistant "Oauth three-legged configuration", instead of "Oauth integration". You'll get better answers that way.

Oracle Assistant is constantly improving, so the assistant does get better with time. The more you use the assistant, the more the assistant improves.

You can tell Oracle Assistant:



- "Find" with a keyword to immediately search the product documentation for information.
- "Not helpful", to make a note so that the team can improve Oracle Assistant.

If you want to move the icon a different place on the page, click on it and drag it to a new position.



3

Get Started with Integration Accelerators and Recipes

Accelerators and recipes are pre-assembled solutions that bundle all the resources required by the integrations, such as the connections and lookups, into one package. Use an accelerator or recipe to quickly get started building your first integration.

Topics:

- Learn About Differences Between Accelerators and Recipes
- Find Accelerators and Recipes
- · Get More Details About Accelerators or Recipes
- Install Accelerators and Recipes
- Configure Resources
- Activate All the Integrations in the Accelerator or Recipe
- Uninstall Accelerators and Recipes

Learn About Differences Between Accelerators and Recipes

Accelerators are run-ready business integrations or technical patterns that you can configure and activate. **Recipes** are templates or samples that give you a head start.

Here's a comparison of accelerators and recipes.

Accelerators	Recipes
A business accelerator provides an end-to-end business process or use case (for example, marketing to lead, hire to retire, or concept to launch).	A recipe is a sample or template that gives you a head start.
A technical accelerator provides a common technical solution (for example, sending alerts on failures). They are meant to be called by another integration.	
Managed and supported by the producer	Not supported
Configurable and extendable	Fully editable in the Oracle Integration designer
Upgrades provided by producer	Can't auto-upgrade to new versions
Configurator in Oracle Integration and as native SaaS	Configurator in Oracle Integration
Paid offering (as decided by producer)	Always free

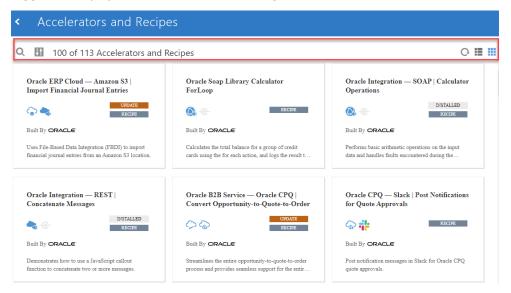


Find Accelerators and Recipes

The **Home** page displays a partial listing of the available accelerators and recipes. It also provides access to the Accelerators and Recipes page, where you can search through the entire portfolio of pre-assembled solutions.

To find an accelerator or a recipe:

- On the Home page, scroll to the Accelerators & Recipes section.
- Click Search All. The Accelerators and Recipes page opens. Note that you can toggle the display between a list view and a grid view.



- 3. Use the options in the toolbar to narrow your search, filter and sort the list, update the list, and change how the list is displayed. You can:
 - Search the list by any text string
 - If the list by type (all, recipe, accelerator) or by provider, and sort the list by name or last updated
 - Refresh the list
 - Switch between a list view or a grid view

Get More Details About Accelerators or Recipes

The Accelerators and Recipes page displays useful information to help you decide whether you want to use a particular accelerator or recipe.

At a glance, you can see which applications the integration connects to, what's the purpose of the integration, what adapters are used, who built it, what type of package it is (ACCELERATOR or RECIPE), and whether it's already installed.





You can also get other details, such as the package name, version number, and publication date, as well as a link to documentation.

To see more information, hover over the card, then click **Open Details** to expand the information pane.

Install Accelerators and Recipes

When you install an accelerator or a recipe, Oracle Integration deploys all the resources required by the integrations in the package. These resources include the connections, certificates (PGP keys), lookups, and libraries.



Oracle Integration comes with the Resequence Messages technical accelerator automatically installed. Therefore, this accelerator will already be listed as **INSTALLED** on the Accelerators and Recipes page when you sign in.

To install accelerators and recipes:

- 1. Find the accelerator or recipe that you want to install.
- 2. Hover over the accelerator or recipe card, then **Install** +.

Configure Resources

Accelerators and recipes are a type of **package**. After you install an accelerator or a recipe, you need to configure all the dependent resources used by all the integrations in the package.

Dependent resources consist of connections, certificates (PGP keys), lookups, and libraries.

- You can edit connections, certificates (PGP keys), lookups, and libraries. You must have administrator privileges to edit certificates.
- You can replace connections and certificates in some packages. However, you can't replace them if an integration in the package is either active or locked. You can't replace them if they are included in a business or technical accelerator.



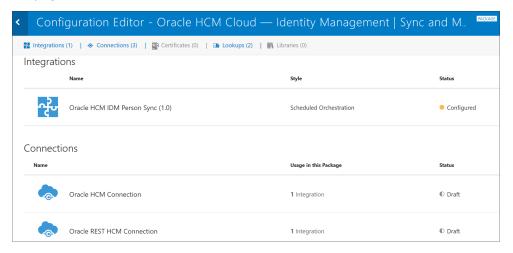
- You can update integration property values for any integration in which properties have been defined.
- You can add schedules to any integration that uses a Scheduled Orchestration style (also called pattern). These integrations run according to the schedule you define.

Editing a dependent resource impacts all integrations that use that resource. Replacing a dependent resource only impacts the integrations in the specific package.

To configure the resources for an accelerator or a recipe:

- 1. Find the accelerator or recipe that you want to configure.
- 2. Hover over the accelerator or recipe card, then click **Configure** .

The Configuration Editor page is displayed. The title bar includes the name of the accelerator or recipe, and indicates that you're working in a package. You can see how many **Integrations**, **Connections**, **Certificates**, **Lookups**, and **Libraries** are included in the package, and click the corresponding link to locate that resource on the page.



Position the cursor in a row to reveal the options for configuring, editing, or replacing a resource.

Click	Valid for	Description
Schedule	Integrations	Define when the integration runs. Use this option only if the integration uses a Scheduled Orchestration style.
Update Property Values	Integrations	Update integration property values for any integration in which properties have been defined.



Click	Valid for	Description
Edit	Connections Certificates Lookups Libraries	Access the corresponding page for editing the resource. For more information about editing these resources: See Configure Connection Properties. See Manage Security Certificates. See Map Data and Create Lookups. See Use Libraries to Manage Functions for Integrations.
Replace	Connections Certificates	Replace a connection or a certificate (PGP key) across all the integrations in a package. • You can replace a connection resource only with another resource of the same role (trigger, invoke, or trigger and invoke). In addition, the connection status must be Configured. You can't replace a connection with a connection whose status is Draft. If a compatible connection resource doesn't exist, a message is displayed. • You can replace a PGP key only with a key of the same type of key. For example, you can replace a public PGP key with another public PGP key, but not with a private PGP key.
Revert	Connections Certificates	Restore the original connection or certificate (PGP key). If you replace a connection or certificate (PGP key) resource, then REPLACED displays after the name of the resource, and the Revert is available.
Open Details	Connections Certificates Lookups Libraries	Display the number and name of the integrations in this package that use the selected resource.

- 4. Make the necessary edits and save your changes. Click the **Go back** icon to return to the Configuration Editor page.
- 5. Continue to edit each resource. When you are ready, return to the Accelerators and Recipes page to activate the integrations in the package.



View Details About a Resource

Whenever you're editing accelerator or recipe packages, you can view which resources are used by which integrations in the package. The resources are the connections, certificates, lookups, and libraries that the integrations depend on to run.

To open the details for a resource:

In the Configuration Editor, hover over a resource and click Open Details ...



The row expands and displays the following information:

- The total number of integrations in the package that use the resource
- The name of each integration in the package that uses the resource

Activate All the Integrations in the Accelerator or Recipe

Accelerators and recipes include any of the resources (connections, certificates, lookups, and libraries) used by any of the integrations in the package. After you configure the connection settings and other resources, you can activate the integrations.

To activate the integrations in an accelerator or a recipe:

- Find the accelerator or recipe that you want to work with.
- Hover over the accelerator or recipe card, then click **Activate** .
- In the Configuration Editor, review the connections, certificates (PGP keys), lookups, and libraries in the package, then click Activate.
- In the Activate Package panel, review messages and select options for the integrations in the package, then click **Activate**.

Uninstall Accelerators and Recipes

Uninstalling an accelerator or a recipe deletes the package and all its integrations.

You can't uninstall accelerators or recipes if there's an integration in the package that has a status of ACTIVE or LOCKED.

Be careful when uninstalling a recipe. You can edit the integrations in a recipe. Note that you'll lose your changes when the integrations are deleted during the uninstall process. Although you can re-install the recipe, the integrations will be the original ones, not your modified ones.

You can't edit the integrations in an accelerator. In this case, you can re-install to restore an accelerator package.

To uninstall an accelerator or a recipe:

- Find the accelerator or recipe that you want to uninstall.
- Hover over the accelerator or recipe card and click **Uninstall** .
- Select which version you want to uninstall, then click Uninstall.



A

Recipes Reference

Oracle Integration includes a portfolio of recipes for integrating different applications that deal with customer relationship management and customer experience (CRM/CX), enterprise resource planning (ERP), human capital management (HCM), social networking, and so on.

Think of a recipe as a template or pre-assembled solution that gives you a head start in building your integrations. You can start with a recipe, and then customize it to fit your needs and requirements.

Topics:

- Cloud Storage/Content Management Recipes
- Customer Relationship Management (CRM) Recipes
- Enterprise Messaging Recipes
- Enterprise Resource Planning (ERP) Recipes
- · Human Capital Management (HCM) Recipes
- Industries Recipes
- Oracle Integration Starter Recipes
- Social and Productivity Recipes
- Other Recipes

Cloud Storage/Content Management Recipes

The Cloud Storage and Cloud Content Management recipes help you integrate applications related to storing and managing data in the cloud.

Topics:

Export Messages from Apache Kafka to Amazon S3

Export Messages from Apache Kafka to Amazon S3

Use this recipe to export Apache Kafka messages to an Amazon Simple Storage Service (Amazon S3) bucket.



This recipe is available as **Apache Kafka** — **Amazon S3 | Export Messages** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe exports messages from an Apache Kafka topic to an Amazon S3 bucket as JSON files. Each Apache Kafka message is exported as a JSON file that contains the contents of the message. The recipe uses the standard Apache Kafka Adapter and the standard REST Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the recipe package. Subsequently, you can activate and run the integration flow manually or specify an execution schedule for it. When triggered, the integration flow queries for messages in the specified Apache Kafka topic and loads predefined number of messages. Then the integration iterates over each message and exports them as JSON files into the specified Amazon S3 bucket.

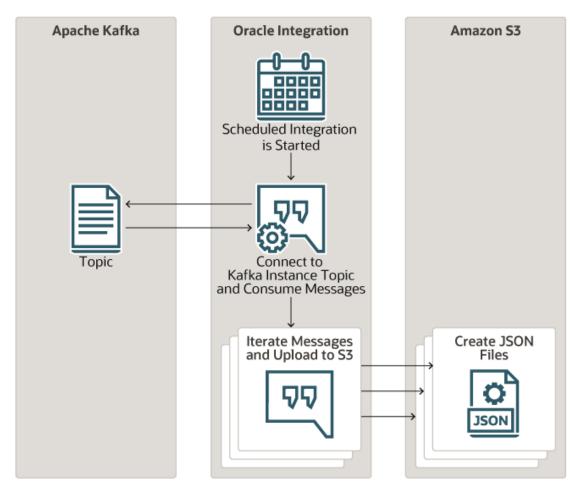
System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Amazon Web Services (AWS) with Amazon S3
- Apache Kafka
- Confluent
- An account in AWS with administrator role
- An account in Confluent with 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 Apache Kafka instance's topic for messages. If messages are present in the specified topic, the integration fetches a predefined number of messages. It iterates over the messages and exports them into Amazon S3. For each exported message a corresponding JSON file that contains the contents of the message is created in the specified Amazon S3 bucket.

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- · Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Apache Kafka and Amazon S3 instances in order to connect to these external systems using Oracle Integration and successfully export messages from Apache Kafka into an Amazon S3 bucket.

- 1. Configure Apache Kafka
- 2. Configure Amazon S3



Configure Apache Kafka

Complete the following configuration tasks to successfully connect to Apache Kafka from Oracle Integration and export messages.

The steps below assume that you're using Confluent as your Apache Kafka instance host.

- 1. Log in to Confluent at https://confluent.cloud.
- 2. In the Confluent Environments page, click the environment you want to use.
- 3. In the resulting page, select the cluster that you want to use.
- 4. Get the Apache Kafka API credentials.
 - a. Expand **Data integration** on the left navigation pane, and click **API keys**.
 - b. In the API keys page, click **Add key**.
 - c. In the Create key window, select Global access and then click Next.
 - The API credentials are displayed on the resulting page.
 - **d.** Note the API credentials, that is the API Key and Secret. You will need these values while configuring the Apache Kafka connection in Oracle Integration.
 - Click the Copy key icon next to Key and note down the API key.
 - Click the **Copy key** icon next to **Secret** and note down the Secret.
 - e. Select the check box I have saved my API key and secret and am ready to continue.
 - f. Click Save.
- Get the Apache Kafka Cluster Bootstrap Server URL.
 - In the left navigation pane, expand Cluster overview and then click Cluster settings.
 - In the Identification section, copy and note the Bootstrap server value.
 You'll need this value while configuring the Apache Kafka connection in Oracle Integration.
- 6. Obtain the truststore file and password, and the keystore file and password. For details, see step 2 in Configure Kafka Confluent Support.
 - You'll need these values while configuring the Apache Kafka connection in Oracle Integration.

In addition to the above configuration tasks, you have to ensure that you create a topic in your Apache Kafka instance to store messages or events. For detail information, see the Apache Kafka Quickstart (https://kafka.apache.org/quickstart).

Configure Amazon S3

Complete the following configuration tasks to successfully connect to Amazon S3 from Oracle Integration and export Apache Kafka messages into an Amazon S3 bucket.

- Log in to your AWS Management Console at https://console.aws.amazon.com/.
- 2. Get your AWS access keys.



- a. In the top navigation bar, choose your user name, and then click **My Security** Credentials from the menu.
- **b.** On the Security Credentials page, expand the Access keys (access key ID and secret access key) section.
- c. Create an access key by clicking Create New Access Key.
 - If the **Create New Access Key** button is disabled, delete one of the existing access keys and try again.
- d. Click **Download Key File** to download the access key ID and secret access key, and save it as a .CSV file to a secure location on your computer.
- 3. Set up your Amazon S3 bucket where you want the Apache Kafka messages to be exported as JSON files.
 - a. In the AWS Management Console, enter S3 in the search field at the top.
 - b. Select S3 (Scalable Storage in the Cloud) from the search results.

The Amazon S3 page appears.

- c. Select an existing bucket.
 - Scroll to the Buckets section. All available buckets are listed under it.
 - Select a bucket.
- d. Alternatively, create and choose a new bucket.
 - Click Create bucket.
 - In the Create bucket dialog, enter a unique name for the bucket, for example oracle-kafka.
 - In the **AWS Region** field, select a region from the drop-down list.
 - Click Create bucket. The new bucket is listed under the Buckets section.
 - Select the bucket.
- 4. Note the **AWS Region** of the bucket. You will need this information when you configure the REST Amazon S3 connection from Oracle Integration.
 - Select the bucket.
 - Click the **Properties** tab, and note the **AWS Region** of the bucket.

Install and Configure the Recipe

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

- 1. On the Oracle Integration Home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.

The list of available recipes appears.

A confirmation message informing that the recipe was successfully installed appears, and the recipe now displays the **Installed** tag.

3. Select the recipe you want to install, and click Install $\stackrel{+}{-}$.



- 4. Click **Configure** to configure the following resources using the **Configuration Editor**.
 - Configure the Oracle Apache Kafka Connection
 - Configure the Oracle REST Amazon S3 Connection
 - Configure the Integration

Configure the Oracle Apache Kafka Connection

You have to configure the connection details for the Oracle Apache Kafka connection.

- 1. On the Configuration Editor page, select Oracle Apache Kafka Connection.
- 2. Click Edit .

The Oracle Apache Kafka Connection window appears.

3. In the **Bootstrap Servers** field, enter the Bootstrap Server URL you obtained earlier while configuring Apache Kafka.

See Configure Apache Kafka.

- 4. In the Security Policy field, select SASL PLAIN Over SSL.
- 5. In the **SASL Username** and **SASL Password** fields, enter the Apache Kafka API Key and Secret that you obtained earlier while configuring Apache Kafka.

See Configure Apache Kafka.

- 6. Complete the following:
 - a. Upload the truststore file in the **TrustStore** field.
 - b. Upload the keystore file in the **KeyStore** field.
 - c. Enter the truststore password in the **Truststore password** field.
 - d. Enter the keystore passowrd in the **KeyStore password** field.

See Configure Apache Kafka.

- 7. Click Save.
- Click Test to ensure that your connection is successfully configured. In the dialog that displays, click Test again. You get a confirmation message if your test is successful.
- 9. Click **Back** to return to the Configuration Editor page.

Configure the Oracle REST Amazon S3 Connection

You have to configure the connection details for the Oracle REST Amazon S3 connection.

- 1. On the Configuration Editor page, select Oracle REST AmazonS3 Connection.
- 2. Click Edit

The Oracle REST AmazonS3 Connection window appears.

3. Under the Connection Properties section, do the following:



- a. In the Connection Type field, select REST API Base URL from the drop-down list.
- b. In the Connection URL field, enter https://s3.<your-bucketregion>.amazonaws.com/. Where your-bucket-region is the region in which you had set up your Amazon S3 bucket.

For example, https://s3.eu-central-1.amazonaws.com/.

See Configure Amazon S3.

- 4. In the Security section, do the following:
 - In the Security Policy field, select AWS Signature Version 4 from the drop-down list.
 - b. Enter the access key and the secret key that you obtained earlier while configuring Amazon S3 in the Access Key and Secret Key fields.
 - c. In the AWS Region field, select the AWS region where you have created the Amazon S3 bucket. For example, eu-central-1.
 - d. In the Service Name field, select Amazon Simple Storage Service (Amazon S3).
 See Configure Amazon S3.
- Click Save.
- 6. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 7. Click **Back** to return to the Configuration Editor page.

Configure the Integration

You have to configure the Apache Kafka adapter endpoint properties in the integration.

- 1. On the Configuration Editor page, select the Oracle Kafka Amazon S3 Scheduled JSON Object Export integration.
- 2. Open the integration in the Integration editor.
- 3. In the integration canvas, select the LoadTopicMessages invoke, and then click the **Edit** icon.

The Apache Kafka Endpoint Configuration Wizard with its Basic Info page appears.

- 4. Click **Next** in the Basic Info page.
- 5. In the Operation page that appears, click **Next** again.

The Topic & Partition page appears.

- Configure the following in the Topic & Partition page.
 - a. In the **Select a Topic** list, choose the Apache Kafka topic from which you want to query and fetch messages.
 - b. Select a partition from the **Specify the Partition** list, or use the **Default** selection.
 - Specify the consumer group to use in the Consumer Group field.
 - d. Select the Read latest option under Specify the option for consuming messages.
 - e. In the Maximum Number of Records to be fetched field, specify the number of messages to be read.
 - f. In the **Do you want to specify the message structure?** field, select **No**.



- g. In the Do you want to specify the headers for the message? field, select No.
- h. Click Next and then click Done.

See Add the Kafka Adapter Connection to an Integration in *Using the Apache Kafka Adapter with Oracle Integration*.

- **7.** On the Integration editor, click **Save**.
- 8. Click **Close** to return back to the Configuration Editor.

Activate and Run the Recipe

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



Before activating and running the recipe, ensure that there are some messages in the Apache Kafka topic.

- 1. Activate the recipe.
 - a. On the Configuration Editor page, click Activate.
 - **b.** On the Activate Package dialog, click **Activate**.

You get a confirmation message that the integration is activated.

- 2. Run the recipe.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Submit Now.

The Schedule Parameters window appears.

- c. In the Schedule Parameters window, specify a value for the AmazonS3BucketName parameter. Enter the name of the Amazon S3 bucket to which you want to export Apache Kafka messages in the **New Value** field. For example, oracle-kafka.
- d. Click Submit.

You've successfully submitted the integration for a test run.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. On the Oracle Integration navigation pane, click Home, then Monitoring, then Integrations, and then Tracking.
 - b. On the Track Instances page, you can see the integration being triggered and executed successfully. The recipe now fetches predefined number of messages from the specified Apache Kafka topic and exports them as JSON files into the Amazon S3 bucket.
- Check if the Apache Kafka messages are exported into the Amazon S3 bucket as JSON files.
 - Log in to your AWS Management Console, and navigate to the Amazon S3 page.



- b. Select the Amazon S3 bucket from the Buckets section of the page.
- c. In the bucket details page, check if there are JSON files for the corresponding Apache Kafka messages.

You can open one of the JSON files using a text editor such as Notepad++ and compare the content of the file with the corresponding Apache Kafka message to see if it is exported correctly.

Related Documentation

- Using the Apache Kafka Adapter with Oracle Integration
- Using the REST Adapter with Oracle Integration

Customer Relationship Management (CRM) Recipes

Oracle Integration includes a set of recipes that help you integrate applications related to managing the relationship with your customers. These recipes handle common business tasks such as converting opportunities to quotes to orders, dispatching service technicians, managing membership, and synchronizing customer data.

Topics:

- Attach Files to Salesforce Records
- Create and Retrieve Organization Details from Oracle Service Cloud (RightNow)
- Create Box Folders for ServiceNow Incidents
- Create Box Folders for Zendesk Organizations
- Create Orders in Oracle B2B Service for Oracle CPQ Opportunities
- Create Salesforce Cases for Jira Issues
- · Create Zendesk Tickets for Jira Issues
- Export Invoices from PayPal to an FTP Server
- Import Marketo Leads to Oracle Autonomous Data Warehouse
- Integrate Oracle Internet of Things Intelligent Applications Cloud with Oracle Field Service Cloud
- Manage ServiceNow Incidents
- Post Slack Notifications for New Marketo Leads
- Post Slack Notifications for ServiceNow Activities
- Send Compliance Documents from DocuSign to New ServiceNow Users
- Send Documents from Salesforce to DocuSign
- Send Notifications to Salesforce Lead Owners Using Twilio
- Send SurveyMonkey Surveys to Salesforce Contacts by Email
- Sync Customer Data Between Oracle Loyalty Cloud and Oracle Responsys
- Sync Data Between Oracle CPQ Transactions and Salesforce Quotes
- Sync Incidents and Resources Between Oracle Field Service Cloud and Oracle B2C Service
- Sync Orgs and Contacts Between Oracle B2C Service and Oracle Marketing Cloud



Attach Files to Salesforce Records

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



This recipe is available as **REST** — **Salesforce | Attach Files to Records** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

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 package and configure the connection and other resources within the package. Subsequently, you can activate the integration flow of the package 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.



You can attach more than one file at a time with this recipe.

System and Access Requirements

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

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- · Activate and Run the Recipe



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.



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.

- 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:
 - 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:
 - 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**.
- 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.



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 package to deploy and configure the integration and associated resources.

On the Oracle Integration home page, scroll to the Accelerators & Recipes section.



- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- **I.** Select the package, and then click the **Install** $\stackrel{+}{}$ icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST Trigger Connection
- Configure the Oracle Salesforce Connection

Configure the Oracle REST Trigger Connection

Test and save the Oracle REST Trigger Connection.

- 1. On the Configuration Editor page, select **Oracle REST Trigger Connection**.
- 2. Click Edit 🕗.
- 3. On the Oracle REST Trigger Connection page, click **Test** to ensure that your connection is correctly configured.

A confirmation message is displayed if your test is successful.

- 4. Click Save. If prompted, click Save for a second time.
- 5. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle Salesforce Connection

- 1. On the Configuration Editor page, select Oracle Salesforce Connection.
- 2. Click Edit .

The connection configuration page appears.

3. In the Connection 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.

4. 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.	



Field	Information to Enter	
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.	

- 5. Click Save. If prompted, click Save for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click Test again.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Activate and Run the Recipe

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

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click Activate again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- Run the recipe from an external application.
 - a. On the Configuration Editor page, select the integration and click **Run** .



- b. On the resulting pop-up dialog, click the **Metadata URL** link.
- c. From the new tab 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. On the Configuration Editor page, select the integration.
 - b. Click Run , then click Test.

The page to test the integration with a sample file is displayed.

c. In the Request section of the test page, 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.

d. Click Test.

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.

- 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.



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
- Using the REST Adapter with Oracle Integration

Create and Retrieve Organization Details from Oracle Service Cloud (RightNow)

Use this recipe to retrieve an organization from Oracle Service Cloud (RightNow) using the SOAP adapter.



This recipe is available as **Create and Retrieve Org Details from Oracle Service Cloud (RightNow)** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

The integration in this recipe uses the SOAP adapter to retrieve an organization from Oracle Service Cloud (RightNow) based on its ID.

The SOAP adapter request contains the ID of the organization to retrieve.

The response contains the ID and the address of the organization. The address includes the street, city, postal code, and country.

The integration that results from installing this recipe is named: Oracle SOAP ServiceCloud Get Organization.

System and Access Requirements

- Oracle Integration
- Oracle Service Cloud (RightNow)

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Before you run the integration created with this recipe, you must configure the connections to the SOAP service and Oracle Service Cloud (RightNow):

- To create a connection to the SOAP service, follow the steps described in Create a SOAP Adapter Connection in Using the SOAP Adapter with Oracle Integration.
- To create a connection to Oracle Service Cloud (RightNow), follow the steps described in Create an Oracle Service Cloud (RightNow) Adapter Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration*.

Related Documentation

To learn more about the adapters used in this recipe see:

- Using the SOAP Adapter with Oracle Integration
- Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration

Create Box Folders for ServiceNow Incidents

Use this recipe to create a folder in Box when an incident is created in ServiceNow.



This recipe is available as **ServiceNow** — **Box | Create Folders for Incidents** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

This recipe creates respective folders in Box for incidents created in ServiceNow. Each ServiceNow incident has a unique system generated number. The name of the folder that gets created in Box for the ServiceNow incident matches the ServiceNow incident's unique number. Also, the Id of the folder created in Box gets updated in the custom field created in ServiceNow.

The recipe uses the standard ServiceNow Adapter and the standard Box Adapter. To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate and run the recipe manually. When the integration flow gets triggered by an incident created in ServiceNow, it checks if a folder with the same name as the newly created incident's unique number already exists in Box. If a similar named folder exists, it ignores the folder creation. Otherwise, it creates a new folder in Box under the specified parent folder, and also creates a shared link for the new folder.

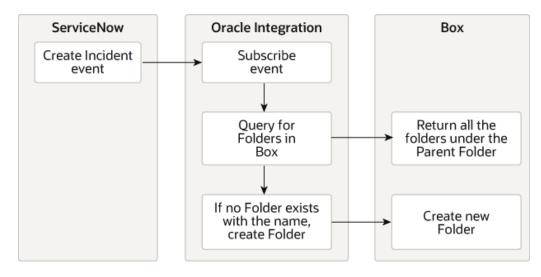
System and Access Requirements

- Oracle Integration Version 21.1.2.0.0 (210129.2200.39447) or higher
- ServiceNow
- Box
- An account in ServiceNow with the Admistrator role
- An account in Box with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.

The integration flow of the recipe is triggered when an incident is created in the ServiceNow instance. The integration then queries for folders under the parent folder in the Box instance. After retrieving all existing folders under the parent folder, if it finds that there is no folder with the same name as the newly created incident's unique number, then it creates a corresponding folder for the incident under the parent folder in Box.





Topics:

- Before You Install the Recipe
- · Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your ServiceNow instance and Box instance in order to connect to these external systems using Oracle Integration and successfully create folders in Box for ServiceNow incidents.

- 1. Configure ServiceNow
- 2. Configure Box

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, then 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*.

You must also create a custom field to store the Id of the Box folder that gets created for new ServiceNow incident. See Create a Custom Field for Box Folder IDs.

Create a Custom Field for Box Folder IDs

This recipe uses unique IDs associated with folders in Box to create Box folders for ServiceNow incidents. Create a custom field for ServiceNow incidents to hold the Box folder IDs.

- 1. Log in to your ServiceNow | Developer instance.
- 2. On the left navigation pane, enter Incident in the Filter navigator field.

The Self Service menu displays all modules related to *Incident*.

- 3. In the Self Service menu, click Incidents.
- In the Incidents window, click New.
- 5. From the available options in the **Incident** menu, click **Configure** and then click **Table**.
- 6. Fill in the required fields, such as Name and Label.
- 7. Click the **Columns** tab, and then click **New** to create a new table column.
- B. In the **Type** field, search for String, and select **String** from the search results.
- 9. Enter the following details:
 - a. In the Column label field, enter the column label. For example, box Id.
 - b. In the Column name field, enter the column name. For example, u box id.
 - c. In the Max length field, enter a value. For example, 255.



10. Click Submit.

Configure Box

You must complete the following prerequisite tasks to connect to Box from Oracle Integration, and to create folders for new ServiceNow incidents under the right parent folder in Box.

- 1. Create and Configure a Box App
- 2. Create a Parent Folder in Box
- 3. Get the Parent Folder ID

Create and Configure a Box App

You have to create an app on the Box platform and configure it to successfully connect to Box from Oracle Integration.

- 1. Log in to the Box Developer console at https://developer.box.com.
- 2. Create a custom app.
 - a. Click Create New App.
 - **b.** In the resulting page to choose the application type, select **Custom App**.
 - c. In the Custom App dialog, select the option User Authentication (OAuth 2.0).
 - d. Enter a unique name for your app in the App Name field.
 - e. Click Create App.

The Box app's configuration page appears.

- Get the OAuth 2.0 credentials.
 - In the Box app's configuration page, scroll to the OAuth 2.0 Credentials section.
 - Copy the client ID from the Client ID field, and the client secret from the Client Secret field.
 - c. Note the Client ID and Secret. You'll need these later while configuring the Box connection from Oracle Integration.
- 4. Specify the redirect URL.
 - a. In the Box app's configuration page, scroll to the OAuth 2.0 Redirect URI section.
 - b. In the **Redirect URI** field, enter the OAuth callback URL of your Oracle Integration instance. Your redirect URI should have the following syntax:

https://Oracle-Integration-instance's-FQDN:443/icsapis/agent/oauth/callback

where, Oracle-Integration-instance's-FQDN is the fully qualified domain name of your Oracle Integration instance, for example, oicinstance.example.com.

- Get the Signature Keys.
 - a. Click the **Webhooks** tab on the top of the page.



- b. Click **Generate Key** to generate the primary and the secondary keys.
- c. Copy the primary and the secondary keys.
- **d.** Note the values of the primary and secondary keys. You'll need these later while configuring the Box connection from Oracle Integration.

For more information about signature keys, see Get Signature Keys in *Using the Box Adapter with Oracle Integration*.

Create a Parent Folder in Box

You must create a parent folder in Box. All folders for new ServiceNow incidents will be created under this parent folder.

- 1. Log in to your Box account at https://account.box.com/login.
- 2. Click **New** on the top right side of the All Files page.
- 3. From the menu that displays, click Folder.
- 4. In the Create a New Folder dialog, enter a suitable name in the Folder Name field.
- 5. Optionally, if you want other people to access the folder, enter their email addresses in the **Invite Additional People** field. Use a comma between each email address.

To control other people's level of access, select the appropriate role (co-owner, editor, viewer, and so on) from the drop-down list in the **Permission** field. Click **Learn More** next to the field to know about the roles and their capabilities.

6. Click Create.

The folder gets created and is displayed on the All Files page.

Get the Parent Folder ID

You have to get the ID of the parent folder in Box, and pass it to the integration properties in Oracle Integration, so that folders for new ServiceNow incidents are created under the right parent folder.

- 1. Log in to your Box account at https://account.box.com/login.
- 2. On the All Files page of your Box instance, search and locate the parent folder that you created.
- 3. Hover your mouse on the parent folder.
- 4. Note the link that appears below on the left side of the page. For example, https://app.box.com/folder/135666632045
- 5. The number after the last forward slash (/) in the link is the folder ID. Note the folder ID. You'll use this later while updating the integration properties in Oracle Integration.

Install and Configure the Recipe

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

- 1. On the Oracle Integration Home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.

The list of available recipes appears.



3. Select the recipe you want to install, and click **Install** +.

A confirmation message informing that the recipe was successfully installed appears, and the recipe now displays the **Installed** tag.

- 4. Click **Configure** on the recipe card to configure the following resources using the **Configuration Editor**.
 - Configure the Oracle ServiceNow Connection
 - Configure the Oracle Box Connection
 - Update Integration Properties

Configure the Oracle ServiceNow Connection

You must configure connection properties and log in credentials for your ServiceNow instance.

- 1. On the Configuration Editor page, select Oracle ServiceNow Connection.
- 2. Click Edit

The ServiceNow Connection window appears.

3. In the Connection Properties section, enter your ServiceNow instance URL in the ServiceNow Instance Name field. For example, https://
<instance name>.service-now.com/.

- **4.** In the Security section, enter your ServiceNow username in the **Username** field, and your ServiceNow password in the **Password** field.
- 5. Click Save.
- Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 7. Click **Back** to return to the Configuration Editor page.

Configure the Oracle Box Connection

You must specify security details such as credentials, permissions, and signature keys to access your Box instance endpoint.

- 1. On the Configuration Editor page, select Oracle Box Connection.
- 2. Click Edit

The Box Connection window appears.

In the Client ID and Client Secret field, enter the client ID and secret you obtained earlier while creating and configuring a Box App.

See Create and Configure a Box App.

- 4. Enter the following authorization permissions in the **Scope** field.
 - root readwrite
 - manage webhook



The root_readwrite permission lets you perform file/folder operations. The manage_webhook permission lets you use webhook operations. For more information on these scopes see the documentation on the Box.com website.

5. In the **Primary Signature Key** and the **Secondary Signature Key** fields, enter the primary and secondary signature keys that you obtained earlier while creating and configuring a Box app.

See Create and Configure a Box App.

- Click Provide Consent.
- 7. 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 Box appears.

- 8. Enter your Box account credentials, and click Authorize.
- 9. In the next page, click Grant access to Box.

You're informed that access to Box is allowed. You can now switch back to the Box Connection window in Oracle Integration to save and test your connection.

- 10. On the Box Connection window, click Save.
- 11. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 12. Click **Back** to return to the Configuration Editor page.

Update Integration Properties

You must update the properties for the integration so that new folders for ServiceNow incidents are created under the right parent folder in Box, and runtime exception emails are sent to the right email ID.

- On the Configuration Editor page, select the Oracle ServiceNow Box Incidents BoxFolder Create integration.
- 2. Click Actions (=), and then click Update Property Values.
- In the Update Property Values dialog, note the two properties that have been configured -NotificationMail and ParentFolderID.
- 4. 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.
- 5. Click **ParentFolderID** and in the **New Value** field, enter the parent folder ID you obtained earlier.

See Get the Parent Folder ID.

6. Click Submit.

You get a confirmation message that the integration properties have been updated successfully.

Activate and Run the Recipe

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



- 1. On the Configuration Editor page, click Activate.
- 2. On the Activate Package dialog, click Activate.

You get a confirmation message that the integration is activated.

- On the Configuration Editor page, select the Oracle ServiceNow Box Incidents BoxFolder Create integration, and click Run
- 4. In the resulting dialog, note the WSDL URL.

You'll use this for configuring basic authentication in ServiceNow to invoke the integration in Oracle Integration from ServiceNow.

Complete the post activation task of invoking the integration, and then test the recipe.

- Invoke the Integration from ServiceNow
- · Test the Recipe

Invoke the Integration from ServiceNow

After activating the recipe, you must configure basic authentication to invoke the integration from ServiceNow.

Configure basic authentication in your ServiceNow instance with the WSDL generated automatically while activating and running the recipe. This invokes the integration in Oracle Integration from ServiceNow.

For detail information on how to configure basic authentication in ServiceNow, see Configure Basic Authentication in *Using the ServiceNow Adapter with Oracle Integration*.

Test the Recipe

Test the recipe to check if a new folder is created in Box when an incident is created in ServiceNow.

- Create an incident in ServiceNow.
 - a. Log in to your ServiceNow | Developer instance as a system administrator.
 - b. On the left navigation pane, enter Incident in the Filter navigator field.

The Self-Service menu displays all modules related to *Incident*.

- c. In the Self-Service menu, click Incidents.
- d. In the Incidents window, click New.
- e. Fill the fields as required in the incident creation form.

Note the unique system generated incident number in the **Number** field of the incident form. For example, INC001234. The name of the folder that gets created in Box in step 3 must be same as this number.

f. Click Submit.

The new incident is created and displayed in the Incident window.

- Monitor the execution of the integration flow in Oracle Integration.
 - a. In the Oracle Integration home page, click **Monitoring**, then **Integrations**, and then **Tracking**.



- b. In the Track instances page, you can see the integration flow being triggered and executed successfully. The recipe now creates a folder in Box under the parent folder whose folder ID we had specified in the integration property value.
- 3. Check if a new folder is created under the Box parent folder.
 - a. Log in to your Box instance.
 - **b.** Search and navigate to the parent folder that you created earlier.
 - c. Check if a new folder is created under the parent folder. The folder name should match with the newly created ServiceNow incident's number. For example, INC001234.

Note that a shared link is also automatically created for the new folder. Click the link icon on the folder's row to open the Shared Link window and view the link.

Related Documentation

- Using the ServiceNow Adapter with Oracle Integration
- Using the Box Adapter with Oracle Integration

Create Box Folders for Zendesk Organizations

Use this recipe to create corresponding folders in Box for all Zendesk organizations.



This recipe is available as **Zendesk** — **Box | Create Folders for Organizations** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe creates a folder in Box for each Zendesk organization as per a schedule specified in Oracle Integration. It uses the standard REST Adapter and Box Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate and run the integration flow of the package manually or specify an execution schedule for it. When triggered, the integration flow queries the Zendesk instance for organization records, retrieves the names of the existing records, and creates a separate folder for each organization record on your Box instance, under a specified parent folder; the folders created shall have the same names as the Zendesk organization names. If a folder for a Zendesk organization already exists in the specified Box location, the integration flow skips the folder-creation activity for that organization record.



This recipe uses only the names of Zendesk organization records to create folders in Box; it doesn't synchronize other data of the records between the applications.



System and Access Requirements

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

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Zendesk and Box instances in order to successfully connect to these external systems using Oracle Integration and create folders in Box for Zendesk organizations.

- Configure Zendesk
- 2. Configure Box

Configure Zendesk

You can create a user account on Zendesk exclusively for integration purposes. To avoid access-related issues, assign the administrator role to this account.

You can use the credentials of this integration user account while configuring the REST-based Zendesk connection in Oracle Integration. For information on creating a new Zendesk user account with the administrator role, see Adding agents and admins.

Configure Box

To access Box from Oracle Integration and create folders for Zendesk organizations, you must perform the following configurations on your Box instance.

- Create an OAuth application and obtain access keys.
 - a. Log in to your Box instance.
 - b. In the left navigation pane, click **Dev Console**.
 - c. In the My Apps page, click Create New App, and then select Custom App.
 - d. In the Custom App dialog:
 - i. Select User Authentication (OAuth 2.0) as the authentication method.
 - ii. Enter a unique name for the app in the **App Name** field.
 - iii. Click Create App.

The application is created and it's Configuration page is displayed.

e. On the application's Configuration page:



- Scroll to the OAuth 2.0 Credentials section and note down the Client ID and Client Secret of the application for later use.
- ii. In the OAuth 2.0 Redirect URI section, enter the OAuth-callback URL of your Oracle Integration instance. Your redirect URI should have the following syntax: https://oracle-Integration-instance's-FQDN:443/icsapis/agent/oauth/callback.
 - where Oracle-Integration-instance's-FQDN is the fully-qualified domain name of your Oracle Integration instance, for example, oicinstance.example.com.
- iii. In the Application Scopes section, select the Write all files and folders stored in Box check box.
- iv. Click Save Changes.
- 2. Create a parent folder to hold all the folders for Zendesk organizations.
 - a. On your Box instance, click **New** in the All Files page.
 - **b.** From the menu that displays, click **Folder**.
 - c. In the Create a New Folder dialog:
 - i. Enter a name for the folder.
 - ii. If you want other people to access the folder, enter their email addresses in the Invite Additional People field; use a space to separate email addresses. Additionally, select the access level for the invitees from the Permission field.
 - iii. Click Create.

The folder is created and displayed on the All Files page.

- 3. Obtain the ID of the parent folder.
 - a. On the All Files page, click the folder's name to open the folder.
 - b. In the folder page's URL, note down the number after the last forward slash (/).
 This is the parent folder's ID, which you'll use while configuring the lookup table of the recipe.

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST Zendesk Connection
- Configure the Oracle Box Connection



Configure the Lookup Table

Configure the Oracle REST Zendesk Connection

- 1. On the Configuration Editor page, select Oracle REST Zendesk Connection.
- 2. Click Edit 🕗.

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter your Zendesk URL, for example, https://my.zendesk.com/

4. In the Security section, enter the following details.

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the user name of the integration user account created on Zendesk. See Configure Zendesk.
Password	Enter the password of the integration user account created on Zendesk.

- 5. Click Save. If prompted, click Save for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

 Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Oracle Box Connection

- 1. On the Configuration Editor page, select **Oracle Box Connection**.
- 2. Click Edit

The connection configuration page appears.

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

Field	Information to Enter
Client ID	Enter the client ID of the OAuth application created in Box. See Configure Box.
Client Secret	Enter the client secret of the OAuth application created in Box.
Scope	Enter root_readwrite.

- 4. Click Save. If prompted, click Save for a second time.
- 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 box.com, where you are prompted to authorize the connection to the Box OAuth application created previously.

7. On the Box web page, click **Grant access to Box**.

You're informed that access to Box is allowed. You can now switch back to the Box Connection window of Oracle Integration to test your connection.

8. In the Oracle Box Connection window, click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

9. Click **Back** to return to the Configuration Editor page. Click **Save** if prompted.

Configure the Lookup Table

Edit the lookup table and enter the ID of the parent folder in Box within which all folders for Zendesk organizations must be created.

- 1. On the Configuration Editor page, select the lookup table and click **Edit** .
- Against the ParentFolderId field, enter the ID of the parent folder in Box. See Configure Box.
- 3. Click **Save**. If prompted, click **Save** for a second time.
- 4. Click **Back** to return to the Configuration Editor page.

Activate and Run the Recipe

After you've configured the connections and the lookup table, activate the recipe package and run it.

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- Run the recipe.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Submit Now.
 - c. In the resulting dialog, click Submit Now again. Note that the Ad hoc request radio button is selected by default.

You've now successfully submitted the integration for execution.



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 execution of the integration flow in Oracle Integration.



- a. On the Configuration Editor page of the recipe, select the integration flow.
- b. Click **Run**, then click **Track Instances**.
- c. On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully. The recipe now creates corresponding folders in Box for all Zendesk organizations.
- 4. Log in to your Box instance and check for the new folders.
 - a. On the All Files page, find the parent folder you created earlier.
 - b. Click the folder's name to open the folder.
 You'll find folders having the same names as your Zendesk organizations listed on the page.

Related Documentation

- Using the Box Adapter with Oracle Integration
- Using the REST Adapter with Oracle Integration

Create Orders in Oracle B2B Service for Oracle CPQ Opportunities

Use this recipe to streamline the entire opportunity-to-quote-to-order process by integrating Oracle B2B Service and Oracle Configure, Price, and Quote (Oracle CPQ).



This recipe is available as **Oracle B2B Service** — **Oracle CPQ | Convert Opportunity-to-Quote-to-Order** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe allows users to create quotes and sales orders for an opportunity. It provides a consistent user interface flow that originates in Oracle B2B Service.

Transaction data is passed seamlessly between the two applications, ensuring accuracy and allowing Oracle B2B Service users to take advantage of the on-demand configuration, pricing, and quoting capabilities of Oracle CPQ.

The benefits include:

- Allows users to create quotes with accurate pricing and generate company specific proposals
- Simplifies quoting and reduces duplicate entry for sales reps managing opportunities that require quotes
- Enables sales reps to accurately configure and price complex products in an intuitive and easy to use manner

System and Access Requirements

Oracle B2B Service, Release 9 or later



- Oracle CPQ, 2015 R1 or later
- Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

 Oracle CPQ Cloud-Oracle Sales Cloud Integration through Oracle Integration Cloud Service Implementation Guide

Create Salesforce Cases for Jira Issues

Use this recipe to create corresponding cases in Salesforce for all Jira issues.



This recipe is available as **Jira** — **Salesforce | Create Cases for Issues** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe creates a case in Salesforce for each Jira issue as per a schedule specified in Oracle Integration. It uses the standard Jira Adapter and the Salesforce Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate and run the integration flow of the package manually or specify an execution schedule for it. When triggered, in the first run, the integration flow reads all the issue records from the Jira instance and, in turn, creates corresponding case records in your Salesforce instance. In subsequent runs, it creates cases for new Jira issues and also propagates updates made in earlier Jira issues to the corresponding Salesforce cases. Basic data associated with Jira issues, such as issue type, summary, priority, description, and status are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462) or higher
- Atlassian
- Salesforce
- An account on Atlassian with the Administrator role
- An account on Salesforce with the Administrator role
- At least one Jira project on your Atlassian account. The project(s) can be in any of the Jira products, for example, Jira Work Management or Jira Software
- At least one issue in your Jira project



Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Atlassian and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and create cases in Salesforce for Jira issues.

- 1. Create an API Token in Your Atlassian Account
- 2. Configure Salesforce

Create an API Token in Your Atlassian Account

To access Jira from Oracle Integration and retrieve issue records, you'll require an API token associated with your Atlassian account.

To create an API token:

- Log in to your Atlassian account using the following URL: https://id.atlassian.com.
- 2. On the Start page, click Account Settings.
- 3. Click **Security** on the left navigation pane.
- On the Security page, scroll to the API token section and click the Create and manage API tokens link.
- 5. On the API Tokens page, click **Create API token**.
- In the resulting dialog, enter a label for the new token, and click Create.A new API token is generated.
- 7. Copy the token's value.

Configure Salesforce

To access Salesforce using Oracle Integration and create cases for Jira issues, 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 create a custom field for case records.

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



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. 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.
 - 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:
 - Scroll to the Administrative Permissions section and ensure that the API Enabled check box is selected.
 - 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.
- 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 User 05.
 - 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.
 - 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:
 - i. Scroll to the Build section in the left navigation pane.



- ii. Expand Develop, and then click API.
- b. 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

- c. 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**.
- Create a custom field for case records.

This recipe uses unique IDs associated with issue records in Jira to synchronize Jira issues with cases in Salesforce. Create a custom field for Salesforce case records to hold the Jira IDs.

- a. On the Salesforce Setup page:
 - Scroll to the Build section in the left navigation pane.
 - ii. Expand Customize, then Cases, and then click Fields.
- On the Case Fields page, scroll to the Case Custom Fields & Relationships section and click New.
- c. On the New Custom Field page:
 - Find and select the Text radio button, and click Next.
 - ii. Enter the following details for the new custom field:
 - Enter Jira Issue 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 Case Fields page, you can see the new field added under the Case Custom Fields & Relationships section.

Install and Configure the Recipe

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

1. On the Oracle Integration home page, scroll to the Accelerators & Recipes section.



- Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** $\stackrel{+}{=}$ icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST Jira Connection
- Configure the Oracle Salesforce Connection

Configure the Oracle REST Jira Connection

- 1. On the Configuration Editor page, select **Oracle REST Jira Connection**.
- 2. Click Edit .

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter your Atlassian instance URL, for example, https://myinstance.atlassian.net/.
Enable two way SSL for outbound connections (Optional)	Select No .

4. In the Security section, enter the following details.

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the email ID associated with your Atlassian account.
Password	Enter the API token you created in your Atlassian account. See Create an API Token in Your Atlassian Account.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

Click Back
 to return to the Configuration Editor page. Click Save again if prompted.

Configure the Oracle Salesforce Connection

1. On the Configuration Editor page, select **Oracle Salesforce Connection**.



2. Click Edit

The connection configuration page appears.

3. In the Connection 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 Configure Salesforce.

4. 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 Configure Salesforce.
Password	Enter the password of the account created for Oracle Integration on Salesforce.



To the password, you must also append the security token generated for the same account.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Activate and Run the Recipe

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

 On the Configuration Editor page, click Activate in the title bar. In the Activate Package dialog, click Activate again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.



The recipe's integration flow contains the following property:

 emailNotification: This integration property holds the email address to which notifications of errors in the integration's execution are sent.

Optionally, you can add or update the value for this property after the integration flow has been activated. For the procedure to update integration properties, see Steps 7 to 9 in Override Design-Time Properties in an Integration.

- 2. Run the recipe.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Submit Now.

The Schedule Parameters page is displayed, where you can specify a value for the <code>lastRun</code> parameter. This parameter stores the date and time 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 Jira issues created or updated after the date-time stamp stored as parameter's <code>Current Value</code> 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 <code>New Value</code> field in the format, <code>yyyy-MM-dd HH:mm</code>. Generally, no input is required.

c. Click Submit on the Schedule Parameters page.

You've now successfully submitted the integration for execution.



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 execution of the integration flow in Oracle Integration.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Track Instances.
 - **c.** On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully.

In the first run, the recipe creates corresponding cases in Salesforce for all of your Jira issues. In subsequent runs, it creates cases for new Jira issues and also propagates updates made in earlier Jira issues to the corresponding Salesforce cases.

- Log in to your Salesforce instance and check for the new (or updated) case records.
 - a. On the Salesforce Setup page, click the **Cases** tab.



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

 On the Cases Home page, select All Open Cases in the View field. Click Go if necessary.

The cases created for Jira issues are displayed.

- c. Click a case number to view a case record.
- d. On the case's page, click the **Details** tab to view the associated Jira issue ID.
 Using the values in the **Jira Issue ID** fields of case records, you can verify if all the required cases have been created or updated.

Related Documentation

- Using the Jira Adapter with Oracle Integration
- Using the Salesforce Adapter with Oracle Integration

Create Zendesk Tickets for Jira Issues

Use this recipe to create corresponding tickets in Zendesk for all Jira issues.



This recipe is available as **Jira** — **Zendesk | Create Tickets for Issues** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe creates a ticket in Zendesk for each Jira issue as per a schedule specified in Oracle Integration. It uses the standard REST Adapter and Jira Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate and run the integration flow of the package manually or specify an execution schedule for it. When triggered, in the first run, the integration flow reads all the issue records from the Jira instance and, in turn, creates corresponding ticket records in your Zendesk instance. In subsequent runs, it creates tickets for new Jira issues and also propagates updates made in earlier Jira issues to the corresponding Zendesk tickets. Basic data associated with Jira issues, such as issue type, summary, priority, description, status, and comments are synchronized between the two platforms.



- For this recipe to execute successfully the Description field of Jira issues must not be blank.
- In the second and subsequent runs of the recipe, updates made to the descriptions of Jira issues aren't synchronized.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462) or higher
- Atlassian
- Zendesk
- An account on Atlassian with the Administrator role
- An account on Zendesk with the Administrator role
- At least one Jira project on your Atlassian account the project(s) can be in any
 of the Jira products (for example, Jira Work Management or Jira Software)
- At least one issue in your Jira project

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Atlassian and Zendesk instances in order to successfully connect to these external systems using Oracle Integration and create tickets in Zendesk for Jira issues.

- 1. Create an API Token in Your Atlassian Account
- 2. Configure Zendesk

Create an API Token in Your Atlassian Account

To access Jira from Oracle Integration and retrieve issue records, you'll require an API token associated with your Atlassian account.

To create an API token:

- Log in to your Atlassian account using the following URL: https://id.atlassian.com.
- 2. On the Start page, click Account Settings.
- 3. Click **Security** on the left navigation pane.
- On the Security page, scroll to the API token section and click the Create and manage API tokens link.
- 5. On the API Tokens page, click Create API token.



- In the resulting dialog, enter a label for the new token, and click Create.A new API token is generated.
- Copy the token's value.

Configure Zendesk

You can create a user account on Zendesk exclusively for integration purposes. To avoid access-related issues, assign the administrator role to this account.

You can use the credentials of this integration user account to access Zendesk using Oracle Integration and create tickets for Jira issues. For information on creating a new Zendesk user account with the administrator role, see Adding agents and admins.

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST Jira Connection
- Configure the Oracle REST Zendesk Connection

Note:

Optionally, you can configure the following lookup tables in the package.

- ORACLE-BRT-JIRA_ZENDESK_PRIORITY: This lookup table contains a
 mapping of Jira issue priorities to Zendesk ticket priorities. Edit the table if
 you require to change the preset mappings.
- ORACLE-BRT-JIRA_ZENDESK_TYPE: This lookup table contains a
 mapping of Jira issue types to Zendesk ticket types. Edit the table if you
 require to change the preset mappings.
- ORACLE-BRT-JIRA_ZENDESK_STATUS: This lookup table contains a
 mapping of Jira issue statuses to Zendesk ticket statuses. Edit the table if
 you require to change the preset mappings.

Configure the Oracle REST Jira Connection

1. On the Configuration Editor page, select **Oracle REST Jira Connection**.



2. Click Edit

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter your Atlassian instance URL, for example, https://myinstance.atlassian.net/.
Enable two way SSL for outbound connections (Optional)	Select No.

4. In the Security section, enter the following details.

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the email ID associated with your Atlassian account.
Password	Enter the API token you created in your Atlassian account. See Create an API Token in Your Atlassian Account.

- 5. Click Save. If prompted, click Save for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle REST Zendesk Connection

- 1. On the Configuration Editor page, select Oracle REST Zendesk2 Connection.
- 2. Click Edit .

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter your Zendesk URL, for example, https://my.zendesk.com/.

4. In the Security section, enter the following details.

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the user name of the integration user account created on Zendesk. See Configure Zendesk.



Field	Information to Enter
Password	Enter the password of the integration user account created on Zendesk.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Activate and Run the Recipe

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

 On the Configuration Editor page, click Activate in the title bar. In the Activate Package dialog, click Activate again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.



The recipe's integration flow contains the following property:

• **emailNotification**: This integration property holds the email address to which notifications of errors in the integration's execution are sent.

Optionally, you can add or update the value for this property after the integration flow has been activated. For the procedure to update integration properties, see Steps 6 to 9 in Override Design-Time Properties in an Integration

- 2. Run the recipe.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Submit Now.

The Schedule Parameters page is displayed, where you can specify a value for the lastRun parameter. This parameter stores the date and time 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 Jira issues created or updated after the date-time stamp stored as 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. Generally, no input is required.

c. Click **Submit** on the Schedule Parameters page.

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 execution of the integration flow in Oracle Integration.
 - a. On the Configuration Editor page, select the integration flow.
 - click **Run**, then click **Track Instances**.
 - c. On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully.

In the first run, the recipe creates corresponding tickets in Zendesk for all of your Jira issues. In subsequent runs, it creates tickets for new Jira issues and also propagates updates made in earlier Jira issues to the corresponding Zendesk tickets.

4. Log in to your Zendesk instance and check for the new (or updated) tickets.

In the **Tags** field within a ticket record, you'll find the corresponding Jira issue ID. Using these tags of the ticket records, you can verify if all the required tickets have been created or updated.

Related Documentation

- Using the Jira Adapter with Oracle Integration
- Using the REST Adapter with Oracle Integration

Export Invoices from PayPal to an FTP Server

Use this recipe to export invoices created in PayPal to an FTP server.



This recipe is available as **PayPal** — **FTP | Export Invoices** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe exports invoices created in PayPal as JSON files to an FTP server, according to a schedule specified in Oracle Integration. Each JSON file contains data of a single invoice.

The recipe uses the standard PayPal Adapter and the FTP Adapter. To use the recipe, you must install the recipe and configure resources such as connections and lookups within the recipe. After you activate and run the integration, the integration uses a schedule parameter to filter out already exported invoices, fetches the latest invoices from PayPal, and writes them as JSON files to the specified location on the FTP server.

System and Access Requirements

- Oracle Integration Version 21.2.1.0.0 (210129.2200.39462) or higher
- An account on PayPal (REST API version 2)
- A secure FTP (sFTP) server



An FTP client to access the sFTP server

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- · Activate and Run the Recipe

Before You Install the Recipe

Follow these steps to enable and configure other systems that this recipe needs to run:

- 1. Configure PayPal
- 2. Access Your FTP Server

Configure PayPal

You must perform certain tasks on your PayPal instance in order to successfully connect to it using Oracle Integration and export invoices.

- Get PayPal API Credentials
- Create Invoices

Get PayPal API Credentials

You have to get the PayPal API credentials, that is, the client ID and client secret which you'll use for establishing connections to PayPal from Oracle Integration.

Log in to the PayPal Developer Dashboard with your account username and password.
 PayPal Developer Dashboard URL:

```
https://developer.paypal.com/developer/applications
```

- 2. In the left menu, under DASHBOARD, click My Apps & Credentials.
- 3. In the page that displays, under App Name, click **Default Application**.

The Sandbox API credentials section displays the Sandbox account name, Client ID and Secret. Click the **Show** link under Secret to view the client secret.

4. Note the client ID and client secret. You'll use these later while configuring the PayPal connection in Oracle Integration.

Create Invoices

Create invoices in your PayPal Sandbox account.

- 1. Get your Sandbox business account credentials.
 - Log in to the PayPal Developer Dashboard using your account user name and password.

PayPal Developer Dashboard URL:

https://developer.paypal.com/developer/applications



b. In the left menu, under SANDBOX, click Accounts.

The default (system generated) personal and business accounts are listed.

- c. Select the account of type **Business**, click in the **Manage accounts** column, and then select **View/edit account**.
- In the resulting Account details dialog, note the Email ID and the System Generated Password.
- Log in to your PayPal Sandbox account using the business account credentials (Email ID and Password) obtained in step 1.

PayPal Sandbox URL:

https://www.sandbox.paypal.com/

- 3. In the Home page, select **Invoicing** under **Quick Links**. Optionally, click the **Pay** & **Get Paid** tab on the top of the page, and select **Create & Manage invoices**.
- 4. Click Create Invoice.
- 5. In the Create invoice page, configure the invoice fields.
 - a. In the **Bill to** field, enter the customer's name. For example, John Doe.
 - b. Click the Add link next to Ship to. In the resulting dialog, enter the customer's shipping and billing address. Click Save.
 - c. Add invoice items. Enter the item name, quantity, price, and tax.

Click Add another line item to add more items in the invoice.

Click Send to send the invoice immediately. Optionally, click Save as Draft to save the invoice and send it later.

Access Your FTP Server

Obtain a sFTP 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 a directory (paypal-invoices) on the server to write the invoice files. Note down the path of the directory.

Install and Configure the Recipe

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

- On the Oracle Integration Home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.

The list of available recipes appears.

Select the PayPal — FTP | Export Invoices recipe, and click Install +.
 A confirmation message informing the recipe was successfully installed appears, and the recipe now displays the Installed tag.



- 4. Click **Configure** on the recipe card to configure the following resources using the **Configuration Editor**.
 - Configure the Oracle PayPal Connection
 - Configure the Oracle FTP Read Write Connection
 - Configure the Oracle REST PayPal Connection
 - Configure Lookups

Configure the Oracle PayPal Connection

You must configure connection properties and log in credentials for your PayPal instance.

- 1. On the Configuration Editor page, select Oracle PayPal Connection.
- 2. Click Edit

The PayPal Connection window appears.

- 3. In the Connection Properties section, from the **PayPal Environment** drop-down list, select the environment (**Sandbox**).
- In the Security section, from the Security Policy drop-down list, select Client Credentials.
- In the Client Id and Client Secret fields, enter the Client ID and Secret that you obtained before.

See Get PayPal API Credentials.

- 6. Click **Save**. If prompted, click **Save** for a second time to confirm.
- Click Test to ensure that your connection is successfully configured. In the dialog that displays, click Test again.

You get a confirmation message if your test is successful.

8. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle FTP Read Write Connection

You must configure connection details such as connection properties and log in credentials for your FTP server.

- 1. On the Configuration Editor page, select Oracle FTP Read Write Connection.
- 2. Click Edit

The FTP Connection window appears.

3. In the Connection 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	22
SFTP Connection	Select Yes from the list.

4. 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.

- 5. Click Save. If prompted, click Save for a second time to confirm.
- Click Test to ensure that your connection is successfully configured. In the dialog that displays, click Test again.

You get a confirmation message if your test is successful.

Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Oracle REST PayPal Connection

You must configure connection details such as connection properties and log in credentials for your REST PayPal connection.

- 1. In the Configuration Editor, select Oracle REST PayPal Connection.
- 2. Click Edit

The REST Connection window appears.

- 3. From the Connection Type drop-down list, select REST API Base URL.
- **4.** In the **Connection URL** field, enter the endpoint URL of your PayPal instance. For example: https://api-sample.sandbox.paypal.com/.
- In the Security section, from the Security Policy drop-down list, select Basic Authentication.
- **6.** In the **Username** field enter the Client ID, and in the **Password** field enter the Client Secret that you obtained before.

See Get PayPal API Credentials.

- 7. Click **Save**. If prompted, click **Save** for a second time to confirm.
- 8. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again.

You get a confirmation message if your test is successful.

 Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure Lookups

Configure lookups to associate values used by one application (PayPal) to values used by the other application (FTP).

- 1. On the Configuration Editor page, select the ORACLE-BRT-PP2FTP_ASSETS lookup.
- 2. Click Edit .



The Lookup window appears.

3. In the Lookup table, specify values for FTP DIR, invPrefix, and pageSize.

config_name	config_value
FTP_DIR	The directory path where the invoice files will be uploaded in the FTP server. For example /oracle-int/paypal-invoices.
invPrefix	Prefix of the invoice files. For example, if the file name is detail_INV-NOV2020.JSON then the prefix is detail
pageSize	Number of files that are to be fetched per call from PayPal. For example 1.

4. Click **Save** and then click **Back** to return to the Configuration Editor page.

Activate and Run the Recipe

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

- 1. On the Configuration Editor page, click Activate.
- 2. In the Activate Package dialog, click **Activate**.

You get a confirmation message that the integration has been activated.

3. On the Configuration Editor page, click **Run** on the integration's row, and then select **Submit Now**. In the resulting dialog, click **Submit Now** again.

The Schedule Parameters window appears.

 In the Schedule Parameters window, enter the current date (format: YYYY-MM-DD) in the New Value field. For example, 2021-06-02.

This filters out already exported invoices and ensures that only the latest invoices that were created in PayPal after the last export gets fetched.

5. Click Submit.

You've successfully submitted the integration for a test run.

Log in to the FTP server and check if the invoice JSON files have been exported into the specified directory.

Related Documentation

- Using the PayPal Adapter with Oracle Integration
- Using the FTP Adapter with Oracle Integration



Import Marketo Leads to Oracle Autonomous Data Warehouse

Use this recipe to import campaign leads from Marketo into Oracle Autonomous Data Warehouse (ADW).



This recipe is available as **Marketo** — **Oracle ADW | Import Leads** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe bulk imports campaign leads created in Marketo and writes them into Oracle ADW database. The recipe uses the standard Marketo Adapter and the standard Oracle Autonomous Data Warehouse Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the recipe package. Subsequently, you can activate and run the integration flow manually or specify an execution schedule for it. When triggered, the integration flow uses a schedule parameter to query campaign leads created in Marketo on or after the date specified in the parameter. It extracts the leads in a file that should be formatted as a comma-separated values file, and writes them into database tables in Oracle ADW.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Marketo
- Oracle Autonomous Data Warehouse (ADW), Oracle Database 19c Enterprise Edition
- An Autonomous Database instance created in Oracle Cloud Infrastructure (OCI)
- An account in Marketo with the Administrator role
- An account in Oracle ADW with the Administrator role
- An Oracle ADW table with write access

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- · Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Marketo and Oracle Autonomous Data Warehouse (ADW) instances in order to connect to these external



systems using Oracle Integration and successfully import leads from Marketo into Oracle ADW.

- 1. Configure Marketo
- 2. Configure Oracle Autonomous Data Warehouse

Configure Marketo

Complete certain configuration tasks in your Marketo instance to successfully connect to it from Oracle Integration.

The following steps give an overview of the tasks that you need to perform to obtain information and details for configuring the Marketo connection from Oracle Integration. For detail information on the steps, see Prerequisites for Creating a Connection in *Using the Marketo Adapter with Oracle Integration*.

- Create an API only user role.
- 2. Create a customer service.
- 3. Obtain the Client ID and Client Secret.
- 4. Obtain the Munchkin ID.

Configure Oracle Autonomous Data Warehouse

Complete configuration tasks in your Oracle Autonomous Data Warehouse (ADW) instance to successfully connect to it using Oracle Integration.

The following steps give an overview of the tasks you need to perform to obtain information and details for configuring the ADW connection property and security from Oracle Integration. For detail information on the steps, see Prerequisites for Creating a Connection in *Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration*.

Download the client credentials wallet.

This is a zip file containing the client security credentials. By default the file name is Wallet <databasename>.zip. You can save this file as any file name you want.

2. Get the ADW database service name.

Note that the service name must be same as the one in the tnsnames.ora file in the client credential wallet.

3. Get the ADW wallet password.

You'll be prompted to enter a wallet password while downloading the client credentials wallet. Note the password as you'll need it while configuring the ADW connection security from Oracle Integration.

Install and Configure the Recipe

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

- 1. On the Oracle Integration Home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.

The list of available recipes appears.

3. Find the recipe package that you want to install.



	A confirmation that the recipe was successfully installed appears, and the recipe now bears the Installed tag.
5.	Click Configure on the recipe card to configure the following resources using the Configuration Editor .
	Configure the Oracle Marketo Connection
	Configure the Oracle ADW Connection
	Configure the Integration
Configure the Ora	acle Marketo Connection
Yo	u have to configure the connection details for the Marketo connection.
1.	On the Configuration Editor page, select Oracle Marketo Connection.
2.	Click Edit .
	The Marketo Connection window appears.
3.	Under the Connection Properties section, enter the munchkin ID that you obtained earlier in the Munchkin ID field.
	See Configure Marketo.
4.	Under the Security section, enter the client ID and secret that you obtained earlier in the Client Id and Client Secret field.
	See Configure Marketo.
5.	Click Save.
6.	Click Test to ensure that your connection is successfully configured.

4. Select the recipe, and click the **Install** + icon.

Configure the Oracle ADW Connection

You have to configure the connection details for the Oracle ADW connection.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if

1. On the Configuration Editor page, select Oracle ADW Connection.

You get a confirmation message if your test is successful.

2. Click Edit .

prompted.

The Oracle ADW Connection window appears.

3. Under Connection Properties, in the **Service Name** field, enter the service name of your Oracle ADW instance.

See Configure Oracle Autonomous Data Warehouse.

- 4. Under the Security section, enter the following details:
 - In the Security Policy field, select JDBC Over SSL from the drop-down list.



- In the Wallet field, click Upload to upload the credential wallet file that you obtained earlier.
- Enter the password for your credential wallet file in the **Wallet Password** field.
- Enter your Oracle ADW account username in the Database Service Username field.
- Enter your Oracle ADW account password in the Database Service Password field.

See Configure Oracle Autonomous Data Warehouse.

- 5. Click **Save** on the connection window.
- 6. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 7. Click **Back** to return to the Configuration Editor page.

See Using the Oracle Mapper with Oracle Integration.

Configure the Integration

You can choose to update the default mappings and the property value configured for the integration in order to adapt it to your use case.

Configure Data Mapping

By default the recipe has a number of configured mappings between Marketo campaign leads and database columns in the Oracle ADW database table. If you want some specific fields to be mapped and stored into Oracle ADW, you can configure them in the mapper.

- In the Configuration Editor page, select the integration's row, and then click Edit from the Actions menu.
- On the integration canvas, select the Map to LoadCampaignLeads icon, and then click Edit.
- 3. Use the mapper to drag the required element nodes from *Sources* to element nodes in *Target*.
- 4. Save the mapping and integration.

Update Integration Properties

You can update the integration's default property value so that notification emails about any errors can be sent to the right person.

- 1. In the Configuration Editor page, select the integration's row, and then click **Update**Property Values from the Actions menu.
- 2. In the Update Property Values dialog, click the **ErrorNotificationEmail** property.
- 3. Enter the email of the person whom you want the error notifications to be sent.
- 4. Click Submit.

You get a confirmation message that the property has been updated successfully.



Activate and Run the Recipe

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



Before activating and running the recipe, ensure that there are some campaign leads in the Marketo instance.

- Activate the recipe.
 - a. On the Configuration Editor page, click Activate.
 - b. On the Activate Package dialog, click Activate.
 You get a confirmation message that the integration is activated.
- 2. Run the recipe.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Submit Now.

The Schedule Parameters window appears.

c. In the Schedule Parameters window, specify a value for the <code>leadStartAt</code> parameter. Enter the date from which you want to import campaign leads from Marketo in the **New Value** field, in the format YYYY-MM-DDThh:mm:ssZ. For example, <code>2021-09-27T00:00:002</code>.

By specifying the date in the schedule parameter, you filter out all leads that were created in the Marketo instance before the specified date, and fetch the leads created on or after the specified date.

d. Click Submit.

You've successfully submitted the integration for a test run.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. On the Oracle Integration navigation pane, click **Home**, then **Monitoring**, then **Integrations**, and then **Tracking**.
 - b. On the Track Instances page, you can see the integration being triggered and executed successfully. The recipe now fetches the campaign leads from the Marketo instance and imports them into Oracle ADW.
- 4. Log in to your Oracle ADW instance and check if the campaign leads from Marketo have been imported into the Oracle ADW database table.

Related Documentation

- Using the Marketo Adapter with Oracle Integration
- Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration



Integrate Oracle Internet of Things Intelligent Applications Cloud with Oracle Field Service Cloud

Use this recipe to automate the process of dispatching technicians to jobs based on alerts from Internet of Things (IoT) enabled devices. It integrates Oracle Internet of Things Intelligent Applications Cloud and Oracle Field Service Cloud.



This recipe is available as **Oracle IoT Cloud** — **Oracle Field Service Cloud** | **Dispatch Technicians** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe enables you to dispatch technicians quickly and increase productivity by having activities automatically created, without any manual interventions, from Oracle Internet of Things Intelligent Applications Cloud to Oracle Field Service Cloud. You can also keep better track of incidents and activities by minimizing human errors.

The capabilities include:

- End-to-end integration between Oracle Field Service Cloud and Oracle Internet of Things Intelligent Applications Cloud
- Automatic activity creation based on alerts from IoT enabled devices
- Display IoT device details within Oracle Field Service Cloud

System and Access Requirements

- Oracle Field Service Cloud, Version 18A or later
- Oracle Internet of Things Intelligent Applications Cloud
- Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

Oracle Field Service Cloud / IoTCS Integration using OIC



Manage ServiceNow Incidents

Use these recipes to create, update, or delete incidents in ServiceNow.



These recipes are available in the **REST** — **ServiceNow | Manage Incidents** package in the Integration Store. Oracle provides these recipes as samples only. These recipes are meant only for guidance, and are not warranted to be error-free. No support is provided for these recipes.

Overview

Using the recipes in the REST — ServiceNow | Manage Incidents package, you can:

- Create an Incident in ServiceNow
- Update an Incident in ServiceNow
- Delete an Incident in ServiceNow

To run the ServiceNow recipes, you must configure the Oracle ServiceNow Connection with the details necessary to access ServiceNow. You can then trigger the integrations through a REST request and specify the request parameters. This invokes the ServiceNow adapter to create, update or delete an incident in ServiceNow, as specified in the request.

- For Oracle REST ServiceNow Create Incident, trigger the integration with the REST request and specify incidentdescription as request parameter in the payload. The ServiceNow adapter is subsequently invoked and it *creates* an incident in ServiceNow. The response containing the incident number and the incident Id is returned to you.
- For Oracle REST ServiceNow Update Incident, trigger the integration with the REST request and specify incidentId, incidentState, and incidentUrgency as request parameters in the payload. The ServiceNow adapter is subsequently invoked and it updates the incident details in ServiceNow.
- For Oracle REST ServiceNow Delete Incident, trigger the integration with the REST request and specify incidentId as request parameter in the payload. The ServiceNow adapter is subsequently invoked and it deletes the incident details in ServiceNow.



Integration Name	Description	REST Commands and Example Payloads
Oracle REST ServiceNow Create	Creates an incident in the ServiceNow	REST API Command:
Incident	instance.	POST: https:// <host:port>/ic/api/</host:port>
		integration/v1/flows/rest/ORCL-R-
		REST_SERVIC_CREATE_INCIDE/1.0/create
		Example Request Payload:
		{ "incidentdescription" : "Employee phone number missing" }
		Example Response Payload:
		{
		"serviceNowNr": "INC0010031", "serviceNowSysId":
		"fe6a74e62f503010f129811df699b6e7"
		}



Integration Name	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/v1/flows/rest/ORCL-R- REST_SERVIC_UPDATE_INCIDE/1.0/update</host:port>
		Example Request Payload:
		<pre>{"incidentId":"bec4ce452f603010f12981 1df699b6cc","incidentState":"2","incidentUrgency":"2"}</pre>
		Note the valid values of request parameters incidentState and incidentUrgency:
		incidentState
		• 1 = New
		• 2 = In Progress
		• 3 = On Hold
		6 = Resolved7 = Closed
		8 = Canceled
		incidentUrgency
		• 1 = High
		• 2 = Medium
		• 3 = Low
		Example Response Payload:
		{ "incidentStatus":
		"bec4ce452f603010f129811df699b6cc"
		}



Integration Name	Description	REST Commands and Example Payloads
Oracle REST ServiceNow Delete	Deletes an incident in the ServiceNow	REST API Command:
Incident	instance.	POST: https:// <host:port>/ic/api/</host:port>
		integration/v1/flows/rest/ORCL-R-
		REST_SERVIC_DELETE_INCIDE/1.0/delete
		Example Request Payload:
		{"incidentId":"fe6a74e62f503010f12981 1df699b6e7"}
		Response Payload:
		{
		"status": "1" }

System and Access Requirements

- Oracle Integration
- ServiceNow

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

To use the recipes in the REST — ServiceNow | Manage Incidents package, you must configure the Oracle REST Connection and the Oracle ServiceNow Connection. You only need to configure these connections once. All the recipes in the package use the same connections.

Connection Name	Parameters
Oracle REST Connection	The connection details are already configured for you. Test and save the connection.
Oracle ServiceNow Connection	Connection Properties ServiceNow Instance Name: Enter the ServiceNow instance URL.
	 Security Security Policy: By default, only the Basic Authentication policy is supported.
	 Username: Enter the username to connect to your ServiceNow instance.
	 Password: Enter the password to connect to your ServiceNow instance.
	For more details, see Create a ServiceNow Adapter Connection in Using the ServiceNow Adapter with Oracle Integration.



Related Documentation

Using the ServiceNow Adapter with Oracle Integration

Post Slack Notifications for New Marketo Leads

Use this recipe to post notification messages in Slack for new Marketo leads.



This recipe is available as **Marketo** — **Slack | Post Notifications for New Leads** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

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 package and configure the connections and other resources within the recipe package. 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

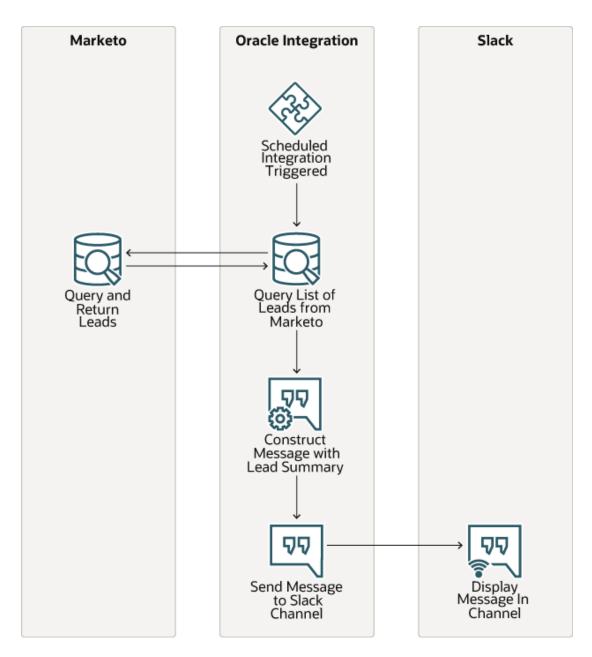
- · Oracle Integration Version 21.2.1 or higher
- Marketo
- Slack
- An account in Marketo with the Administrator role
- An account in Slack with the Administrator role

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.





Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

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.

- 1. Configure Marketo
- 2. Configure Slack

Configure Marketo

Complete the following tasks in your Marketo instance to successfully connect to it from Oracle Integration.

- 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*.

Configure Slack

Complete the following tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

- 1. Create and Configure a Slack App
- 2. Set Up Your Slack WorkSpace and Channel
- 3. Get the Slack Channel ID

Create and Configure a Slack App

You have to create a Slack app, and configure it to receive notifications for new leads in Marketo.

- 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.
 - 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.



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.
 - Scroll to the Scopes section, and under User Token Scopes click Add an OAuth Scope.
 - f. Enter chat:write.
 - a. 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.

- 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** $\stackrel{Q_+}{\longrightarrow}$ on the top right of the channel.
 - 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 package to deploy and configure the integration and associated resources.

- On the Oracle Integration Home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.

The list of available recipes appears.

3. Select the recipe you want to install, and click **Install** +.

A confirmation message informing that the recipe was successfully installed appears, and the recipe now displays the **Installed** tag.

- 4. Click **Configure** to configure the following resources using the **Configuration** Editor.
 - Configure the Oracle Marketo Connection
 - Configure the Oracle Slack Connection
 - Update Integration Property

Configure the Oracle Marketo Connection

You have to configure the connection details for your Marketo connection.

- 1. On the Configuration Editor page, select Oracle Marketo Connection.
- 2. Click Edit

The Marketo Connection window appears.

Under the Connection Properties section, enter the munchkin ID that you obtained earlier in the Munchkin ID field.



See Prerequisites for Creating a Connection in *Using the Marketo Adapter with Oracle Integration*.

 Under the Security section, enter the client ID and secret that you obtained earlier in the Client Id and Client Secret field.

See Prerequisites for Creating a Connection in *Using the Marketo Adapter with Oracle Integration*.

- Click Save.
- 6. Click **Test** to ensure that your connection is successfully configured.

You get a confirmation message if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle Slack Connection

You must configure connection details for your Slack connection.

- 1. On the Configuration Editor page, select Oracle Slack Connection.
- 2. Click Edit .

The Slack Connection window appears.

- 3. Under the Security section:
 - In the Client ID and Client Secret fields, enter the Client ID and Client Secret you
 obtained earlier while creating and configuring a Slack app.
 - In the Scope field, enter chat:write:bot.

See Create and Configure a Slack App.

4. Click Provide Consent.

A new browser window opens to approve access to Slack.

- 5. Click Allow.
- 6. On the Slack Connection window, click **Save**.
- 7. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 8. Click **Back** to return to the Configuration Editor page.

Update Integration Property

You have to update the integration property value so that notification messages about new Marketo leads are sent to the right Slack channel.

- In the Configuration Editor page, select the Oracle Marketo Slack Schd New Lead Notification integration.
- 2. Click Actions , then click Update Property Values.
- 3. In the Update Property Values dialog, click SlackChannelld.
- 4. In the **New Value** field, enter the channel ID of your Slack channel.

See Get the Slack Channel ID.



5. 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.



Before activating and running the recipe, ensure that there are new leads in your Marketo instance.

- 1. On the Configuration Editor page, click Activate.
- 2. On the Activate Package dialog, click Activate.

You get a confirmation message that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- 3. Run the recipe.
 - a. On the Configuration Editor page, select the integration.
 - b. Click Run , then click Submit Now.
 - c. In the resulting dialog, click **Submit Now** again. Note that the **Ad hoc request** radio button is selected by default.

You've now successfully submitted the integration for execution.



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 execution of the integration flow in Oracle Integration.
 - Return to the Oracle Integration home page. In the left navigation pane, click
 Monitoring, then Integrations, and then Tracking.
 - b. On the Track Instances page, you'll see the integration flow of the recipe being triggered and executed successfully. The recipe now posts a notification message in the specified Slack channel.
- 5. Check if a notification message is posted in Slack.
 - 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



Using the Slack Adapter with Oracle Integration

Post Slack Notifications for ServiceNow Activities

Use this recipe to post notification messages in Slack for IT service management (ITSM) activities such as asset creation, incident creation, request creation, or request closure in ServiceNow.

Note:

This recipe is available as **ServiceNow** — **Slack | Post Activity Notifications** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe posts notification messages in Slack when certain activities occur in ServiceNow. The recipe contains the following four integration flows:

- Oracle ServiceNow Slack Asset Creation: This integration flow is triggered when an
 asset is created in ServiceNow. It receives the asset details (such as DisplayName,
 Category, Created By, Created Date, SerialNumber) from ServiceNow, and posts the
 same to the specific channel in Slack.
- Oracle ServiceNow Slack Incident Creation: This integration flow is triggered when an
 incident is created in ServiceNow. It receives the incident details (such as
 IncidentNumber, Created By, Description, Priority, Assigned To, Creation Date) from
 ServiceNow, and posts the same to the specific channel in Slack.
- Oracle ServiceNow Slack Request Creation: This integration flow is triggered when a
 request is created in ServiceNow. It receives the request details (such as Request
 Number, Description, Priority, Created By, Created Date) from ServiceNow, and posts the
 same to the specific channel in Slack.
- Oracle ServiceNow Slack Request Closure: This integration is triggered when a
 request is closed in ServiceNow. It receives details (such as Request Number, Priority,
 Created By, Created Date, Closed By) of the closed request from ServiceNow, and posts
 the same to the specific channel in Slack.

The integrations use the standard ServiceNow Adapter and the standard Slack Adapter. To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Then, you can activate and run some or all the integrations in the package, and invoke the integrations from ServiceNow.

System and Access Requirements

- Oracle Integration version 21.2.1.0.0 (210129.2200.39462) or higher
- ServiceNow
- Slack, Production 4.15.0 64-bit
- An account on ServiceNow with the Administrator role
- An account on Slack



Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

Follow these steps to enable and configure other systems that this recipe needs to run:

- 1. Configure ServiceNow
- 2. Configure Slack

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, then 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*.

Configure Slack

You must perform certain tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

- 1. Set Up Your Slack WorkSpace and Channel
- 2. Get the Slack Channel ID
- 3. Create and Configure a Slack App

Set Up Your Slack WorkSpace and Channel

You must set up a workspace in Slack, create channels in the workspace, and then add people who should receive the specific notification messages into the channels.

- 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, #incident-creation.
- d. Optionally, enter a suitable description in the **Description** field to let others know what the channel is about. For example: This channel notifies whenever an incident is created in ServiceNow.
- e. Click Create.

Repeat the steps above to create four channels.

- A channel to post ServiceNow asset creation notications (#asset-creation).
- A channel to post ServiceNow incident creation notications (#incident-creation).
- A channel to post ServiceNow request creation notications (#request-creation).
- A channel to post ServiceNow request closure notications (#request-closure).
- 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** $\stackrel{Q_+}{=}$ 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.

Repeat the steps above for the four channels.

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.
- 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.

Repeat the steps above for all the four Slack channels to get their Channel IDs.

Create and Configure a Slack App

You have to create a Slack app, and configure it to receive notifications sent by Oracle Integration when an asset, incident, or request is created, and when a request is closed in ServiceNow.

- 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**.
- 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.
 - 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.

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 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
 - · 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.

g. Press Enter.



Install and Configure the Recipe

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

- 1. On the Oracle Integration Home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.

The list of available recipes appears.

- 3. Find and select the recipe package that you want to install, then click **Install** +.

 A confirmation message informing that the recipe was successfully installed appears, and the recipe now displays the **Installed** tag.
- 4. Click **Configure** on the recipe card to configure the following resources using the **Configuration Editor**.
 - Configure the Oracle ServiceNow Connection
 - Configure the Oracle Slack Connection
 - Configure Integration Properties

Configure the Oracle ServiceNow Connection

You must configure connection properties and log in credentials for your ServiceNow instance.

- 1. On the Configuration Editor page, select Oracle ServiceNow Connection.
- 2. Click Edit .

The ServiceNow Connection window appears.

- 3. In the Connection Properties section, enter your ServiceNow instance URL in the ServiceNow Instance Name field. For example, https://<instance_name>.servicenow.com
- 4. In the Security section, enter your ServiceNow username in the **Username** field, and your ServiceNow password in the **Password** field.
- 5. Click Save.
- 6. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 7. Click **Back** to return to the Configuration Editor page.

Note that since the connection is used by multiple integrations in the recipe package, you may be prompted in a Save and Exit dialog to save your changes before you exit the connection window. Click **Save**.

Configure the Oracle Slack Connection

You must configure connection details for your Slack instance.

1. On the Configuration Editor page, select Oracle Slack Connection.



2. Click Edit

The Slack Connection window appears.

- 3. Under the Security section:
 - In the **Client ID** and **Client Secret** fields, enter the Client ID and Client Secret you obtained earlier while creating and configuring a Slack app.
 - In the Scope field, enter the permission scopes you configured while creating and configuring a Slack app.

See Create and Configure a Slack App.

4. Click Provide Consent.

A new browser window opens to approve access to Slack.

- 5. Click Allow.
- 6. On the Slack Connection window, click Save.
- Click Test to ensure that your connection is successfully configured. In the dialog that displays, click Test again. You get a confirmation message if your test is successful.
- 8. Click **Back** to return to the Configuration Editor page.

Note that since the connection is used by multiple integrations in the recipe package, you may be prompted in a Save and Exit dialog to save your changes before you exit the connection window. Click **Save**.

Configure Integration Properties

You must configure the integration properties for each of the integrations in the recipe package so that notifications are sent to the right channel and runtime exception emails are sent to the right email ID.

- 1. On the Configuration Editor page, select the integration.
 - For updating properties for ServiceNow asset creation notifications, select the Oracle ServiceNow Slack Asset Creation integration.
 - For updating properties for ServiceNow incident creation notifications, select the Oracle ServiceNow Slack Incident Creation integration.
 - For updating properties for ServiceNow request creation notifications, select the Oracle ServiceNow Slack Request Creation integration.
 - For updating properties for ServiceNow request closure notifications, select the Oracle ServiceNow Slack Request Closure integration.
- 2. Click Actions , and then select Update Property Values from the menu.
- In the Update Property Values dialog, note the two properties that have been configured - SlackChannelID and EmailNotification.
- Click SlackChannelID and in the New Value field enter the channel ID of your Slack channel.

For example, if you're configuring the properties for Oracle ServiceNow Slack Asset Creation integration, then enter the channel ID of the Slack channel you created to post notifications for ServiceNow asset creation (#asset-creation).



See Get the Slack Channel ID.

- 5. Click **EmailNotification** and in the **New Value** field enter the email ID where you want to notify in case a runtime exception/error occurs while running the integration.
- 6. Click Submit.

You get a confirmation message that the properties have been updated.

Activate and Run the Recipe

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

- 1. On the Configuration Editor page, click **Activate**.
- 2. In the Activate Package dialog, click Activate.

You get a confirmation message that the integration is activated.

- 3. On the Configuration Editor page, select the integration you want to run, and click **Run**
- 4. In the resulting dialog, note the WSDL URL.

You'll use this for configuring basic authentication in ServiceNow to invoke the integration in Oracle Integration from ServiceNow.

Complete the post activation task of invoking the integration, and then test the recipe.

- Invoke the Integration from ServiceNow
- Test the Oracle ServiceNow Slack Asset Creation Recipe
- Test the Oracle ServiceNow Slack Incident Creation Recipe
- Test the Oracle ServiceNow Slack Request Creation Recipe
- Test the Oracle ServiceNow Slack Request Closure Recipe

Invoke the Integration from ServiceNow

After activating the recipe, you must configure basic authentication to invoke the integration from ServiceNow.

Configure basic authentication in your ServiceNow instance with the WSDL generated automatically while activating and running the recipe. This invokes the integration in Oracle Integration from ServiceNow.

For detail information on how to configure basic authentication in ServiceNow, see Configure Basic Authentication in *Using the ServiceNow Adapter with Oracle Integration*.

Test the Oracle ServiceNow Slack Asset Creation Recipe

Test the recipe to check if a notification is sent to Slack when an asset is created in ServiceNow.

Before testing the recipe, check if you've run the **Oracle ServiceNow Slack Asset Creation** integration in Oracle Integration.

- Create an asset in ServiceNow.
 - a. Log in to your ServiceNow | Developer instance as a system administrator.



b. On the left navigation pane, enter Asset in the Filter navigator field.

The Self-Service menu displays all modules related to Asset.

- c. In the Self-Service menu, under **Portfolios** click **All Assets**.
- d. In the Assets window, click New.
- e. In the resulting window, select the asset type.
- f. Fill the fields as required in the asset creation form.
- g. Click Submit.

The new asset is created and displayed in the Assets window.

- 2. Monitor the execution of the integration flow in Oracle Integration.
 - a. In the Oracle Integration home page, click **Monitoring**, then **Integrations**, and then **Tracking**.
 - b. In the Track instances page, you can see the integration flow being triggered and executed successfully. The recipe now sends a notification message to the Slack channel created for asset creation notifications.
- 3. Check if the notification is posted in the Slack channel.
 - In Slack, navigate to the channel (#asset creation) created for posting asset creation notifications.
 - **b.** Check if the notification message for the asset that you created in ServiceNow is posted in the channel.

Test the Oracle ServiceNow Slack Incident Creation Recipe

Test the recipe to check if a notification is sent to Slack when an incident is created in ServiceNow.

Before testing the recipe, check if you've run the **Oracle ServiceNow Slack Incident Creation** integration in Oracle Integration.

- Create an incident in ServiceNow.
 - a. Log in to your ServiceNow | Developer instance as a system administrator.
 - b. On the left navigation pane, enter Incident in the Filter navigator field.

The Self-Service menu displays all modules related to *Incident*.

- c. In the Self-Service menu, click Incidents.
- d. In the Incidents window, click New.
- e. Fill the fields as required in the incident creation form.
- f. Click Submit.

The new incident is created and displayed in the Incident window.

- 2. Monitor the execution of the integration flow in Oracle Integration.
 - **a.** In the Oracle Integration home page, click **Monitoring**, then **Integrations**, and then **Tracking**.
 - b. In the Track instances page, you can see the integration flow being triggered and executed successfully. The recipe now sends a notification message to the Slack channel created for incident creation notifications.



- 3. Check if the notification is posted in the Slack channel.
 - In Slack, navigate to the channel (#incident creation) created for posting incident creation notifications.
 - **b.** Check if the notification message for the incident that you created in ServiceNow is posted in the channel.

Test the Oracle ServiceNow Slack Request Creation Recipe

Test the recipe to check if a notification is sent to Slack when a request is created in ServiceNow.

Before testing the recipe, check if you've run the **Oracle ServiceNow Slack Request Creation** integration in Oracle Integration.

- Create an asset in ServiceNow.
 - a. Log in to your ServiceNow | Developer instance as a system administrator.
 - $\textbf{b.} \quad \text{On the left navigation pane, enter $\tt Request$ in the Filter navigator field.}$
 - The Self-Service menu displays all modules related to Request.
 - c. Scroll down the Self-Service menu, and under **Service Catalog** click **Requests**.
 - d. In the Service Catalog window, select a category. If required, select a sub-category.
 - e. Fill the fields as required in the request creation form.
 - f. Click Submit.
- 2. Monitor the execution of the integration flow in Oracle Integration.
 - a. In the Oracle Integration home page, click Monitoring, then Integrations, and then Tracking.
 - b. In the Track instances page, you can see the integration flow being triggered and executed successfully. The recipe now sends a notification message to the Slack channel created for request creation notifications.
- 3. Check if the notification is posted in the Slack channel.
 - **a.** In Slack, navigate to the channel (#request-creation) created for posting request creation notifications.
 - **b.** Check if the notification message for the request that you created in ServiceNow is posted in the channel.

Test the Oracle ServiceNow Slack Request Closure Recipe

Test the recipe to check if a notification is sent to Slack when a request is closed in ServiceNow.

Before testing the recipe, check if you've run the **Oracle ServiceNow Slack Request Closure** integration in Oracle Integration.

- 1. Create an asset in ServiceNow.
 - a. Log in to your ServiceNow | Developer instance as a system administrator.
 - b. On the left navigation pane, enter Request in the Filter navigator field.
 - The Self-Service menu displays all modules related to Request.
 - c. Scroll down the Self-Service menu, and under **Service Catalog**, and then **Open Records**, click **Requests**.



The Request window opens listing all open requests.

- d. Select the request you want to close from the list.
- e. In the Approval field, select either Approved or Rejected.
- f. In the **Request state** field, select an option from the drop-down list.
- g. Click Update.
- 2. Monitor the execution of the integration flow in Oracle Integration.
 - a. In the Oracle Integration home page, click **Monitoring**, then **Integrations**, and then **Tracking**.
 - b. In the Track instances page, you can see the integration flow being triggered and executed successfully. The recipe now sends a notification message to the Slack channel created for request creation notifications.
- 3. Check if the notification is posted in the Slack channel.
 - **a.** In Slack, navigate to the channel (#request closure) created for posting request closure notifications.
 - **b.** Check if the notification message for the request that you closed in ServiceNow is posted in the channel.

Related Documentation

- Using the ServiceNow Adapter with Oracle Integration
- Using the Slack Adapter with Oracle Integration

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.



This recipe is available as **ServiceNow** — **DocuSign | Send IT Compliance Documents** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be errorfree. No support is provided for this recipe.

Overview

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 package. 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

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your ServiceNow instance and DocuSign instance inorder to connect to these external systems using Oracle Integration and successfully send compliance documents from DocuSign to new ServiceNow users.

- Configure ServiceNow
- 2. Configure DocuSign

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, then 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*.

Configure DocuSign

You must complete the following prerequisite tasks to connect to DocuSign from Oracle Integration, and send compliance documents for new ServiceNow users.

- 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.
- 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.
- 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.
 - 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.
- On the DocuSign home page, click Templates on the title bar.
- 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.
- 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.

- 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 package to deploy and configure the integration and associated resources.

- On the Oracle Integration Home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.

The list of available recipes appears.

Select the recipe you want to install, and click Install +.
 A confirmation message informing the recipe was successfully installed appears, and the recipe now displays the Installed tag.

- 4. Click **Configure** to configure the following resources using the **Configuration Editor**:
 - Configure the Oracle ServiceNow Connection
 - Configure the Oracle DocuSign Connection
 - Update Integration Properties

Configure the Oracle ServiceNow Connection

You must configure connection properties and log in credentials for your ServiceNow instance.

- 1. On the Configuration Editor page, select Oracle ServiceNow Connection.
- 2. Click Edit

The ServiceNow Connection window appears.

3. In the Connection Properties section, enter your ServiceNow instance URL in the ServiceNow Instance Name field. For example, https://
<instance name>.service-now.com/.

- 4. In the Security section, enter your ServiceNow username in the **Username** field, and your ServiceNow password in the **Password** field.
- 5. Click Save.
- Click Test to ensure that your connection is successfully configured. In the dialog that displays, click Test again. You get a confirmation message if your test is successful.
- 7. Click **Back** to return to the Configuration Editor page.

Configure the Oracle DocuSign Connection

You must specify security details such as the integrator key (Client ID), client secret and account ID to access your DocuSign instance endpoint.

1. On the Configuration Editor page, select Oracle DocuSign Connection.



2. Click Edit

The DocuSign Connection window appears.

 In the Client ID (Integrator Key) and Client Secret fields, enter the client ID (integration key) and secret that you obtained earlier while creating an app and integration key in DocuSign.

See Create an App and Integration Key.

- 4. Enter signature extended in the Scope field.
- **5.** Enter sandbox in the Instance Type field.
- 6. In the **Account ID** field, enter the DocuSign Account ID followed by the API Account ID.

You obtained the DocuSign Account ID and the API Account ID while creating an app and integration key in DocuSign. See Create an App and Integration Key.

- Click Provide Consent.
- 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.

- 9. Enter your DocuSign account credentials, then click **Authorize** and grant access.
- 10. On the DocuSign Connection window, click Save.
- 11. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 12. Click **Back** to return to the Configuration Editor page.

Update Integration Properties

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.

- On the Configuration Editor page, select the Oracle ServiceNow DocuSign User Doc SignIn integration.
- 2. Click Actions , and then click Update Property Values.
- 3. In the Update Property Values dialog, note the three properties that have been configured DocuSignAccountId, DocuSignTemplateId, and NotificationMail
- 4. 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.

Click DocuSignTemplateId, and in the New Value field, enter the Template ID that you obtained earlier.

See Get the Template ID.

- 6. 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.
- 7. Click Submit.



You get a confirmation message that the integration properties have been updated successfully.

Activate and Run the Recipe

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

- 1. On the Configuration Editor page, click **Activate**.
- On the Activate Package dialog, click Activate.You get a confirmation message that the integration is activated.
- 3. On the Configuration Editor page, select the Oracle ServiceNow DocuSign User Doc SignIn integration, and click **Run**.
- 4. In the resulting dialog, note the WSDL URL. You'll use this for configuring basic authentication in ServiceNow to invoke the integration in Oracle Integration from ServiceNow.

Complete the post activation task of invoking the integration, and then test the recipe.

- Invoke the Integration from ServiceNow
- Test the Recipe

Invoke the Integration from ServiceNow

After activating the recipe, you must configure basic authentication to invoke the integration from ServiceNow.

Configure basic authentication in your ServiceNow instance with the WSDL generated automatically while activating and running the recipe. This invokes the integration in Oracle Integration from ServiceNow.

For detail information on how to configure basic authentication in ServiceNow, see Configure Basic Authentication in *Using the ServiceNow Adapter with Oracle Integration*.

Test the Recipe

Test the recipe to check that when a new user is created in ServiceNow, the user receives an email to electronically sign a compliance document.

- Create a new user in ServiceNow.
 - a. Log in to your ServiceNow | Developer instance as a system administrator.
 - On the left navigation pane, enter User Administration in the Filter navigator field.

The Self-Service menu displays all modules related to *User Administration*.

- c. In the Self-Service menu, under **User Administration** click **Users**.
- d. In 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.

- 2. Monitor the execution of the integration in Oracle Integration.
 - a. In the Oracle Integration home page, click **Monitoring**, then **Integrations**, and then **Tracking**.
 - b. In the Track instances page, you can see the integration flow being triggered and executed successfully. The recipe now sends the compliance document to the new user's email address that was specified in the new user record form in ServiceNow.
- 3. 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 1e.
 - **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



Using the DocuSign Adapter with Oracle Integration

Send Documents from Salesforce to DocuSign

Use this recipe to send documents attached to a Salesforce opportunity to DocuSign.



This recipe is available as **Salesforce** — **DocuSign | Send Opportunity Documents** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be errorfree. No support is provided for this recipe.

Overview

This recipe sends all the documents attached to a Salesforce opportunity to DocuSign when the opportunity is moved to the *Closed Won* status in Salesforce. It uses the standard Salesforce Adapter and the DocuSign Adapter. To use the recipe, you must install the recipe package and configure the connections and other resources within the package.

When an opportunity record closes with the status *Closed Won* in Salesforce, the integration flow of the recipe is triggered. It receives the opportunity ID from Salesforce and checks if the opportunity record contains any document attachments. If there are attachments, the integration flow retrieves those documents and sends them to DocuSign. In DocuSign, these documents are available within a new draft envelope.

Note:

- The recipe sends a document from Salesforce to DocuSign only when the opportunity record containing the document is moved to the Closed Won status in Salesforce.
- The recipe sends only valid PDF documents from Salesforce to DocuSign.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39447) or higher
- Salesforce
- DocuSign Developer instance
- An account on Salesforce with the Administrator role
- An account on the DocuSign Developer instance with the Administrator role

Topics:

- · Before You Install the Recipe
- Install and Configure the Recipe



· Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and DocuSign instances in order to successfully connect to these external systems using Oracle Integration and transfer documents from Salesforce to DocuSign.

- 1. Configure Salesforce
- 2. Configure DocuSign

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve documents attached to opportunities, you must perform certain configuration tasks 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 recipe in Oracle Integration, and create a workflow rule to send the outbound message when a Salesforce opportunity is moved to the status *Closed Won*. Finally, enable uploading attachments to records as Salesforce Files.

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



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. See Toggle or switch between Lightning Experience and Salesforce Classic.

- 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.
 - 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:
 - 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.



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:
 - Enter a first name and last name for the user, for example, Integration User 05.
 - 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.
 - 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:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Develop**, and then click **API**.
 - b. 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

- c. 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. On your Salesforce instance, create an outbound message to trigger the integration flow of this recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - 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:
 - Select Opportunity in the Object field, and click Next.
 - ii. Enter the following details for the new outbound message:
 - Enter OpportunityDocusign as the name of the 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/v1/flows/salesforce/ORCL-R-SFDC_DOCUSIGN_OPPO_DOCSIG/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 Opportunity fields to send, verify that Id is added to the Selected Fields box.
- 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 opportunity is moved to the status *Closed Won*.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - 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 **Opportunity** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:
 - Enter OpportunityDocusign as the rule's name.
 - Optionally, enter a description for the rule.
 - In the Evaluation Criteria section, verify that the created, and any time it's edited to subsequently meet criteria radio button is selected.
 - In the Rule Criteria section, select **formula evaluates to true** in the first drop-down field, and add the following expression into the expression field: IsWon = True.
 - 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: OpportunityDocusign.
 - 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. Enable uploading attachments to records as Salesforce Files. See Files are uploaded as Salesforce Files and not as Attachments.



You must enable this feature only if you upload documents in the Classic UI through the Notes & Attachments section. In the Lightning UI, the documents you upload in the Notes & Attachments section of records are added as Salesforce Files by default.

Configure DocuSign

To connect to DocuSign from Oracle Integration and send documents to it, you must create an OAuth application on your DocuSign instance and obtain access keys for the application.

- 1. Log in to your DocuSign Developer account at https://appdemo.docusign.com/home.
- 2. Click the **Settings** tab on the title bar.
- 3. On the resulting page, scroll to the INTEGRATIONS section in the left navigation menu, 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 and click **CREATE APP**.
- **6.** On the app's page:
 - a. Under the General Info section, copy the **Integration Key** displayed using the **Copy** to clipboard icon.
 - **b.** In the Authentication section:
 - i. Under User Application, select Authorization Code Grant.
 - ii. Under **Secret Keys**, click **ADD SECRET KEY**. After the secret key is added, copy its value using the **Copy to clipboard** icon.
 - c. In the Additional settings section, under Redirect URIs, click ADD URI and enter the following redirect URI.

```
https://{OIC HOST}:{OIC SSL PORT}/icsapis/agent/oauth/callback
```

Replace OIC_HOST and OIC_SSL_PORT with values specific to your Oracle Integration instance.



d. Click Save.

You're redirected back to the Apps and Keys page. The newly created app is displayed under the Apps and Integration Keys section.

Optionally, on the Apps and Keys page, get the API account ID of your DocuSign account. You'll require this ID if you want to log in to multiple accounts for a given user.

Under the My Account Information section, copy the **API Account ID** displayed using the corresponding **Copy to clipboard** icon.

- 8. Get the DocuSign account ID.
 - a. Click the profile image in the upper-right corner of the page.
 - b. In the menu that displays, note the account ID under your name.

Install and Configure the Recipe

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

- On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle Salesforce Connection
- Configure the Oracle DocuSign Connection
- · Update Integration Properties

Configure the Oracle Salesforce Connection

- 1. On the Configuration Editor page, select Oracle Salesforce Connection.
- 2. Click Edit .

The connection configuration page appears.

3. In the Connection 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 Configure Salesforce.

4. 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 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.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- **6.** Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle DocuSign Connection

- 1. On the Configuration Editor page, select **Oracle DocuSign Connection**.
- 2. Click Edit

The connection configuration page appears.

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

Field	Information to Enter
Client ID	Enter the integration key of the OAuth application created in DocuSign. See Configure DocuSign.
Client Secret	Enter the client secret of the OAuth application created in DocuSign.
Scope	Enter signature extended.
Instance Type	Select Sandbox.
Account ID	Enter the API account ID of your DocuSign account. See Configure DocuSign.

- 4. Click **Save**. If prompted, click **Save** for a second time.
- 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 DocuSign login page.

7. Enter your DocuSign account credentials and click Continue to log in.

You are now displayed a page that prompts you to authorize the connection to the DocuSign application created previously.

8. On the DocuSign web page, click ACCEPT.



You're informed that access to DocuSign is allowed. You can now switch back to the DocuSign Connection window of Oracle Integration to test your connection.

9. In the Oracle DocuSign Connection window, click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

10. Click Back to return to the Configuration Editor page. Click Save if prompted.

Update Integration Properties

Update the integration properties with appropriate values.

The recipe's integration flow contains the following properties:

- DocusignAccountID: This integration property holds the unique number associated with a DocuSign account. Enter the ID of your account. To obtain this ID, see Step 8 in Configure DocuSign.
- **NotificationMail**: This integration property holds the email address to which runtime exception emails are sent. Enter an email address of your choice.

For the procedure to update these integration properties, see Steps 7 to 9 in Override Design-Time Properties in an Integration.

Activate and Run the Recipe

After you've configured the connections and the integration, activate the recipe package and run it.

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- 2. Run the recipe.
 - a. Log in to your Salesforce instance.



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

b. On the Salesforce Setup page, click the **Opportunities** tab.



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

c. On the Opportunities Home page, select All Opportunities in the View field and click Go!.



All the opportunities on your Salesforce instance are displayed.

- Find a sample opportunity record that hasn't been closed, and click its name to open it.
- e. On the opportunity's page:
 - Scroll to the Notes & Attachments section, click Upload Files, and upload a sample PDF document.
 - ii. In the Opportunity Detail section at the top, click **Edit**, then select **Closed Won** in the **Stage** field, and click **Save**.

You've now successfully triggered the recipe.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. On the Configuration Editor page of the recipe, select the integration flow.
 - b. Click Run , then click Track Instances.
 - **c.** On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully.

The recipe now creates a draft envelope in DocuSign with the document attachment of the Salesforce opportunity you just closed.

- **4.** Log in to your DocuSign Developer instance at https://appdemo.docusign.com/home and check for the new draft envelope.
 - a. On the DocuSign instance, click the Manage tab on the title bar.
 - On the resulting page, in the ENVELOPES section in the left navigation menu, click
 Drafts to view the envelope with the Salesforce opportunity's document.

Related Documentation

- Using the DocuSign Adapter with Oracle Integration
- Using the Salesforce Adapter with Oracle Integration

Send Notifications to Salesforce Lead Owners Using Twilio

Use this recipe to send a notification message to a Salesforce lead's new owner when the owner is changed.



This recipe is available as **Salesforce** — **Twilio | Send Notifications to Lead Owners** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe sends a notification message to the newly-assigned owner of a Salesforce lead using Twilio's short message service (SMS). It uses the standard Salesforce Adapter and the Twilio Adapter. To use the recipe, you must install the recipe package and configure the connections and other resources within the package.



When a lead's owner changes in Salesforce, the integration flow of the recipe is triggered. The integration receives the information about the lead along with the new owner's mobile number from Salesforce, creates a text message, and invokes Twilio to send the message to the mobile number.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39447) or higher
- Salesforce
- Twilio
- An account on Salesforce with the Administrator role
- An account on Twilio

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Twilio instances in order to successfully connect to these external systems using Oracle Integration and send messages to Salesforce users.

- 1. Configure Salesforce
- 2. Configure Twilio

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve information about Salesforce leads, you must perform certain configuration tasks 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 recipe in Oracle Integration, and create a workflow rule to send the outbound message when the owner of a Salesforce lead is changed.

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



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. 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:
 - 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:
 - 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.



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:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand Develop, and then click API.
 - b. 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



- c. 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.** On your Salesforce instance, create an outbound message to trigger the integration flow of this recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - 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 LeadTwilio as the name of the 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/v1/flows/ salesforce/ORCL-R-SFDC TWILIO LEAD SEND SMS/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 all the 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 owner is changed.
 - 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 LeadTwilio as 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 add the following expression into the expression field: ISCHANGED (OwnerId).
 - 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: LeadTwilio.
 - 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.

Configure Twilio

To connect to Twilio from Oracle Integration and send messages through it, you must obtain the credentials associated with your Twilio account.

- Log in to the Twilio console at https://console.twilio.com/.
- 2. Under the Project Info section on the home page:
 - **a.** Note the phone number associated with your account. Twilio uses this phone number to send messages.
 - b. Note the values in the ACCOUNT SID and AUTH TOKEN fields. You'll need these values to configure the Twilio connection in Oracle Integration. Click Show to reveal the AUTH TOKEN number.

Install and Configure the Recipe

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

- On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** $\stackrel{+}{}$ icon.



5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle Salesforce Connection
- Configure the Oracle Twilio Connection
- Update Integration Properties

Configure the Oracle Salesforce Connection

- 1. On the Configuration Editor page, select Oracle Salesforce Connection.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection 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 Configure Salesforce.

4. 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 Configure Salesforce.
Password	Enter the password of the account created for Oracle Integration on Salesforce.



To the password, you must also append the security token generated for the same account.

- 5. Click Save. If prompted, click Save for a second time.
- **6.** Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle Twilio Connection

1. On the Configuration Editor page, select **Oracle Twilio Connection**.



2. Click Edit

The connection configuration page appears.

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

Field	Information to Enter
AccountSID	Enter the Account SID of the Twilio account. See Configure Twilio.
AuthToken	Enter the Auth Token of the Twilio account.

- 4. Click Save. If prompted, click Save for a second time.
- Click Test to ensure that your connection is successfully configured.
 A confirmation message is displayed if your test is successful.
- 6. Click **Back** to return to the Configuration Editor page. Click **Save** if prompted.

Update Integration Properties

Update the integration properties with appropriate values.

The recipe's integration flow contains the following properties:

- NotificationMail: This integration property holds the email address to which runtime exception emails are sent. Enter an email address of your choice.
- **FromNumber**: This integration property holds the Twilio phone number that is used to send a message. To obtain this number, see Configure Twilio. Ensure that you enter the mobile phone number and associated country code within double quotes. For example, "+19234567891". If there is a mismatch between the number entered here and the number associated with your Twilio account, Oracle Integration sends a run-time exception email.

For the procedure to update these integration properties, see Steps 7 to 9 in Override Design-Time Properties in an Integration.

Activate and Run the Recipe

After you've configured the connections and the integration, activate the recipe package and run it.

- 1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.
 - A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.
- 2. Run the recipe.
 - Log in to your Salesforce instance.



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.

b. On the Salesforce Setup page, click the **Leads** tab.

Note:

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

- c. On the Leads Home page, select All Open Leads in the View field and click Go!.
 - All open leads on your Salesforce instance are displayed.
- d. Click on a lead record's name to open it.
- e. On the lead's page, click **Change** in the **Lead Owner** field.
- f. On the resulting Change Lead Owner page, with User selected as the owner type, select an existing user using the lookup. Click Save.

Ensure that the Salesforce user you select has a mobile number associated with it. This recipe requires the mobile phone number of the lead owner to send an SMS message. If you don't find the required user in the lookup, create a new user record (with a mobile phone number).

You've now successfully triggered the recipe.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. On the Configuration Editor page of the recipe, select the integration flow.
 - b. Click Run , then click Track Instances.
 - **c.** On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully.

The recipe now fetches information about the lead record you just edited along with the new owner's mobile number from Salesforce and invokes Twilio to send a message to the mobile number.

On the mobile phone of the user you just assigned to a Salesforce lead, check for the SMS message from the Twilio account.

The message contains information about the Salesforce lead assignment.

Related Documentation

- Using the Twilio Adapter with Oracle Integration
- Using the Salesforce Adapter with Oracle Integration



Send SurveyMonkey Surveys to Salesforce Contacts by Email

Use this recipe to email a SurveyMonkey survey to a Salesforce case's contact when the case is closed.



This recipe is available as **Salesforce** — **SurveyMonkey | Send Surveys to Case Contacts** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe sends a link to a survey created with SurveyMonkey to the contact's email address specified in a Salesforce case when the case is closed. It uses the standard Salesforce Adapter and the SurveyMonkey Adapter. To use the recipe, you must install the recipe package and configure the connections and other resources within the package.

When a case is closed in Salesforce, the integration flow of the recipe is triggered. It receives the contact email address specified in the case from Salesforce, fetches a survey link from SurveyMonkey, and sends the link to the email address.

System and Access Requirements

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

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- · Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and SurveyMonkey instances in order to successfully connect to these external systems using Oracle Integration and email the surveys to Salesforce contacts.

- 1. Configure Salesforce
- 2. Configure SurveyMonkey



Configure Salesforce

To access Salesforce using Oracle Integration and retrieve contact email addresses from Salesforce cases, you must perform certain configuration tasks 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 recipe in Oracle Integration, and create a workflow rule to send the outbound message when a Salesforce case is closed.

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



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. See Toggle or switch between Lightning Experience and Salesforce Classic.

- 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.
 - 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:
 - Scroll to the Administrative Permissions section and ensure that the API Enabled check box is selected.
 - 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.



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.
- 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:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand Develop, and then click API.
 - b. 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
```

- c. 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**.
- On your Salesforce instance, create an outbound message to trigger the integration flow of this recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - 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 Case in the Object field, and click Next.
 - ii. Enter the following details for the new outbound message:
 - Enter SurveyMonkeyCase as the name of the 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/v1/flows/
 salesforce/ORCL R SFDC SM CASE CLOSE/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 Case fields to send, select all the fields.
- Click Save.

The new outbound message is created and displayed on your browser.

- Now, create a workflow rule to send the outbound message (created previously) when a Salesforce case is closed.
 - **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 Case in the Object field, and click Next.
 - ii. Enter the following details for the new workflow rule:
 - Enter SurveyMonkeyCase as 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, with the **criteria are met** option selected, select **Case: Closed** from the **Field** drop-down list, **equals** from the **Operator** field, and enter True in the **Value** field.
 - 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: SurveyMonkeyCase.
 - 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.



Configure SurveyMonkey

You must perform the following prerequisite tasks in your SurveyMonkey instance.

- 1. Create and Configure a SurveyMonkey App
- 2. Create a Survey

Create and Configure a SurveyMonkey App

You have to create a SurveyMonkey app and configure it to access SurveyMonkey APIs from Oracle Integration.

- 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. Click **SETTINGS** to open the app's settings, and in the **OAuth Redirect URL** field, enter the URL of your Oracle Integration instance.
- Get the SurveyMonkey API credentials.
 - a. Click **SETTINGS** to open the app's settings.
 - b. 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.

- Select the scopes.
 - a. 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
 - b. Click Update Scopes.

You get a confirmation message that the scopes are successfully updated.



Create a Survey

You have to create a survey in SurveyMonkey for which you want to receive responses.

- Log in to your SurveyMonkey account at https://www.surveymonkey.com/user/ sign-in/.
- 2. On the SurveyMonkey Dashboard, click CREATE SURVEY on the title bar.
- **3.** 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.

- 4. Click **Next** to preview your survey.
- 5. Click **Next** to view the COLLECT RESPONSES section.
- 6. Click Send surveys your way.

The COLLECT RESPONSES section displays the different ways in which you can collect the responses.

7. 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.

Install and Configure the Recipe

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

- On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.



- Configure the Oracle Salesforce Connection
- Configure the Oracle SurveyMonkey Connection
- Update Integration Properties

Configure the Oracle Salesforce Connection

- 1. On the Configuration Editor page, select Oracle Salesforce Connection.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection 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 Configure Salesforce.

4. 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 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.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle SurveyMonkey Connection

- 1. On the Configuration Editor page, select Oracle SurveyMonkey Connection.
- 2. Click Edit

The connection configuration page appears.

3. 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.

- 4. Click **Save**. If prompted, click **Save** for a second time.
- 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 SurveyMonkey login page.

7. 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.

8. 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.

In the Oracle SurveyMonkey Connection window, click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

10. Click **Back** to return to the Configuration Editor page. Click **Save** if prompted.

Update Integration Properties

Update the integration properties with appropriate values.

The recipe's integration flow contains the following properties:

- **SurveyName**: This integration property holds the name of the survey that you have created in SurveyMonkey. To obtain this name, see Create a Survey.
- CollectorName: This integration property holds the name of the web link collector that you have created in SurveyMonkey. To obtain this name, see Create a Survey.
- **EmailNotification**: This integration property holds the email address to which runtime exception emails are sent. Enter an email address of your choice.

For the procedure to update these integration properties, see Steps 7 to 9 in Override Design-Time Properties in an Integration.

Activate and Run the Recipe

After you've configured the connections and the integration, activate the recipe package and run it.

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.



A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- 2. Run the recipe.
 - Log in to your Salesforce instance.



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

b. On the Salesforce Setup page, click the **Cases** tab.



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

c. On the Cases Home page, click **New** to create a new case.

The New Case page is displayed where you can enter details of the case.

- d. On the New Case page:
 - In the Case Information section, select an existing contact using the lookup in the Contact Name field.

Ensure that the contact you select has an email address associated with it. This recipe requires the email address of the case contact to send the survey link.



If you don't find the required contact in the lookup, create a new contact record (with an email address) using the **Contacts** tab.

- ii. Leave New selected in the Status field.
- iii. Select the required value in the Case Origin drop-down list.
- iv. Select the required value in the Case Currency drop-down list.
- v. Click Save.

The new case is created and displayed on your browser.

- e. On the summary page of the case, click the **Details** tab and then click **Close Case**.
- f. On the resulting page, select Closed in the Status field, select a suitable value in the Case Reason field, and click Save.

You've now successfully triggered the recipe.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. On the Configuration Editor page of the recipe, select the integration flow.
 - b. Click Run , then click Track Instances.



c. On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully.

The recipe now fetches the link to the specified survey from SurveyMonkey and sends it to the contact email address of the Salesforce case you just closed.

4. Log in to the email account specified in the Salesforce case you closed, and check for the email message from Oracle Integration.

The email contains the URL of the SurveyMonkey survey.

Related Documentation

- Using the SurveyMonkey Adapter with Oracle Integration
- Using the Salesforce Adapter with Oracle Integration

Sync Customer Data Between Oracle Loyalty Cloud and Oracle Responsys

Use this recipe to promote member engagement and sync customer profile data from Oracle Loyalty Cloud into Oracle Responsys. You can target, influence, and reward your customers by using this integrated loyalty and marketing solution.



This recipe is available as **Oracle Loyalty Cloud** — **Oracle Responsys | Manage Membership** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

Oracle Loyalty Cloud enables you to manage the complete loyalty program life cycle. Oracle Responsys provides your B2C marketing teams with a centralized canvas to build consumer profiles and tailor interactions for consumers across email, mobile, display, and social channels.

With this recipe, you can promote member engagement and sync customer profile data from Oracle Loyalty Cloud into Oracle Responsys. For example, you can sync the following member attributes:

- First name
- Last name
- Address
- Email address
- Mobile phone number

You can change the attributes used in this recipe by editing the integration mapping within Oracle Integration.



With this integration, you can use customer data in Oracle Loyalty Cloud for targeting and personalization in Oracle Responsys Marketing Platform Cloud Service (Oracle Responsys) in the following scenarios:

- Sync customer profile data from Oracle Loyalty Cloud into Oracle Responsys. This synchronization would apply for any customer becoming a new loyalty member or for any changes to the profile of an existing loyalty member.
- Real-time triggering of Oracle Responsys marketing orchestration programs for specific events in Oracle Loyalty Cloud, such as joining a loyalty program, cashing out points, or moving to a new tier. The auto-trigger requires pre-creating orchestration programs in Oracle Responsys that correspond to Oracle Loyalty Cloud events.
- Life cycle marketing for campaigns like birthdays, newsletters, and special promotions for members with specific characteristics. In Oracle Responsys, you define the specific campaign, targeting criteria, and personalization rules, as well as the recurring schedule to run the campaign.

System and Access Requirements

- Oracle Loyalty Cloud
- Oracle Responsys Marketing Platform Cloud Service
- Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

 Oracle Loyalty Cloud Integrating Loyalty with Oracle Responsys Marketing Platform Cloud Service

Sync Data Between Oracle CPQ Transactions and Salesforce Quotes

Use this recipe to synchronize data between Oracle Configure, Price, and Quote (Oracle CPQ) transactions and Salesforce quotes.



This recipe is available as **Oracle CPQ** — **Salesforce | Synchronize Data** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe performs bidirectional synchronization of data between transactions in Oracle CPQ and quotes in Salesforce. The recipe contains two integration flows - one for synchronizing Salesforce quotes with corresponding Oracle CPQ transactions, and another for synchronizing Oracle CPQ transactions with corresponding Salesforce quotes. The integrations use the standard Oracle CPQ Adapter and the Salesforce Adapter.



To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate the integrations. When a transaction is created, updated (for example, a new line item is added), or canceled in Oracle CPQ, the *Oracle CPQ Salesforce Transaction2Quote Sync* integration is triggered and it creates, updates, or sets to denied status the corresponding Salesforce quote. Similarly, when a quote is created, updated (for example, the value of a quote line item is changed), or set to denied status in Salesforce, the *Oracle Salesforce CPQ Quote2Transaction Sync* integration is triggered and it creates, updates, or cancels the corresponding transaction in Oracle CPQ.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462) or higher
- Oracle CPQ, Version 4.15.0 64-bit
- Salesforce
- An account on Oracle CPQ with the Superuser role
- An account on Salesforce with the Administrator role

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle CPQ and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and synchronize data in both systems.

- Configure Oracle CPQ
- Configure Salesforce

Configure Oracle CPQ

To access an Oracle CPQ instance using Oracle Integration and ensure that data is synchronized, you must perform certain configuration tasks on your Oracle CPQ instance.

Log in to your Oracle CPQ instance as an Administrator and complete the following tasks.

- Obtain the SOAP WSDL from the Oracle CPQ site.
 - Oracle Integration uses the Oracle CPQ transaction WSDL to understand the valid data and operations provided by Oracle CPQ.
 - For details, see Prerequisites for Creating a Connection in *Using the Oracle CPQ Adapter with Oracle Integration*.
- Obtain the REST catalog URL of Oracle CPQ.



This URL is used to access the Oracle CPQ application's business resources for outbound connections. It is formatted as $http(s)://site_URL/rest/v3/metadata-catalog$ where site URL is the base URL of the Oracle CPQ site.

For details, see Prerequisites for Creating a Connection in *Using the Oracle CPQ Adapter with Oracle Integration*.

- 3. Create a user account for Oracle Integration with the **Web Services Only** permissions.
 - a. On the Oracle CPQ home page, click **Admin Drawer** on the top left of the title bar.
 - 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 detail information, see Setting Up Users in the Oracle CPQ online help.
- 4. Create a connection from Oracle CPQ to Oracle Integration by creating an integration.
 - a. On the Oracle CPQ home page, click **Admin Drawer** on the top left of the title bar.
 - **b.** In the Admin navigation pane, expand **Integration Platform** and click **Integration Center**.
 - c. On the Integration Center page, click Create Integration.
 - d. From the **Type** drop-down list, select **Integration Cloud Service**.
 - In the resulting page enter the following details, and click **Test** to verify if your connection works.

Information to enter
Enter a suitable name.
This field is automatically populated.
Variable names can only contain alpha- numeric characters and underscores. The entry can be changed before saving, but after saving the value is read-only.
Enter the Oracle Integration instance URL.
Enter the username to access your Oracle Integration instance.
Enter the password to access your Oracle Integration instance.

- f. Select Enable Integration, and then click Save.
- 5. Link the integration that you created to an existing process.
 - a. On the Oracle CPQ home page, click **Admin Drawer** on the top left of the title bar.
 - b. In the Admin navigation pane, expand Commerce and Documents and click Process Definition.
 - **c.** 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**.
 - d. On the resulting Integrations page, click Add.
 - e. On the Select Integration Types page, select Integration Cloud Service and click Next.
 - f. On the Edit Integration page, enter the following details and click **Add**:



Field	Information to Enter
Name	Enter the name of the integration cloud service. For example, Salesforce Quote Update.
Variable Name	Enter a variable name. For example, salesforceQuoteUpdate
Timeout	Enter the timeout in milliseconds. For example, 30000.
Description	If required, enter a suitable description.
Action	Select the Export option.
Services	Select the service. For example, Oracle CPQ Salesforce Transaction2Quote Sync.

- **6.** 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 Action List page, click the Copy Of Save link.
 - d. On the Admin Action page, click the **Integration** tab.
 - e. From the Integration List box, select the integration created in Step a and click the > arrow to add it to the Selected Integration box.
 - f. Click Update.
- Configure the Salesforce ID.

Set up the Salesforce ID as an attribute so that in case you need to update the Oracle CPQ transaction, you can use the Salesforce ID as a lookup to identify the change required.

- 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 **Attributes** from the **Navigation** drop-down list, and click **List**.
- c. On the Attribute List page, scroll down to the end of the page and click Add.
- d. On the Attribute Editor page, enter the following details and click Add:

Field	Information to Enter
Label	Enter the name of the attribute. For example, QuoteID.
Variable Name	Enter the variable name. For example, quoteID.
Attribute Type	Select Text.

- e. In the Text Attribute Editor, enter a suitable description in the **Description** field, and click **Update**.
- 8. Deploy the process.

Now that you've created the integration and configured it with the required action and attribute, you can deploy the process linked to the integration.



- a. On the Processes page, select the process linked to the integration. Select
 Deployment Center from the Navigation drop-down list, and click List.
- On the Deployment Center page, select Deploy under Event Type, and click Add Event.
- c. Click Refresh.

Once the deployment completes, the **Last Deployed** time will be updated.

Configure Salesforce

To access Salesforce using Oracle Integration and ensure that the opportunity quote data is synchronized, you must perform certain configuration tasks on your Salesforce instance.

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



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. See Toggle or switch between Lightning Experience and Salesforce Classic.

- 1. Create an API-enabled custom role.
 - 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**, and on the resulting page complete the following:
 - i. Select Standard User in the Existing Profile field.
 - ii. Enter a name for the new profile, for example, API Enabled.
 - iii. Click Save.
 - **c.** On the Profile Detail page, click **Edit**, and on the resulting Profile Edit page complete the following:
 - 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 complete the following:
 - In the Accounts row, leave Read, Create, Edit, and Delete selected.
 Additionally, select ViewAll.
 - In the Contacts row, leave Read, Create, Edit, and Delete selected.
 Additionally, select ViewAll.
 - In the Price Books row, leave Read selected. Additionally, select Create,
 Edit. and Delete.
 - In the Products row, leave Read selected. Additionally, select Create, Edit, and Delete.
 - iii. Scroll down to the end of the page and click Save.



2. Create a user account for Oracle Integration and assign the custom role created in step 1 to this account.

You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

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.
- In the resulting page, click New User.
- c. Create a username and password.
 - On the New User page, enter the first and last name of the user. For example, Integration User05.
 - ii. In the Email field, enter a valid email address. Note that the email address you enter is automatically populated in the Username field. Note the username.
 - iii. In the User License field, select Salesforce.
 - 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.
 - vi. 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.
 - vii. 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.

- d. 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 from the menu.
 - iii. On the My Settings page, under the Quick Links section, click **Edit my personal information**.



- iv. On the left navigation pane of the resulting page, click Reset My Security Token.
- v. Select 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:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand Develop, and then click API.
 - **b.** 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

- c. 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 ID for the Standard Price Book.
 - a. On the Salesforce Setup page, click the **Products** tab.
 - b. Scroll to the Maintenance section and click Manage Price Books .
 - c. On the resultant Price Books page, click Standard Price Book in the Price Book Name column.

The Standard Price Book details page appears.

d. Locate the 15 character Salesforce ID in the Standard Procice Book page's URL. For example, if the URL is https://<instance_name>.my.salesforce.com/ 01s0o75956AleUQ, then the Salesforce ID is 01s0o75956AleUQ.

Note the Salesforce ID. You'll require it when you configure the lookup table in Oracle Integration.

- 6. Configure the lookup value for Oracle CPQ Cloud.
 - a. In the left navigation pane of the Salesforce Setup page, scroll to the Build section, expand **Customize**, then **Quotes**, and then click **Fields**.
 - **b.** On the resultant Quote Fields page, scroll to the Quote Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page, select the **Text** radio button and click **Next**.
 - d. Enter the following details for the new custom field and click **Next**:



Field	Information to Enter
Field Label	Enter the field name. For example, TransactionId_CPQ.
Length	Enter the maximum length that you want to allow for the text field. For example, 255.
Field Name	This field is automatically populated.
Description	If required, enter a description for the custom field.

- e. On the Establish field-level security page, select Visible and click Next.
- f. Leave all check boxes selected in the next page, and click **Save**.

On the Quote Fields page, you can see the new custom field added under the Quote Custom Fields & Relationships section.

- 7. Configure the Salesforce platform event. The platform event notifies external systems about changes to the Salesforce opportunity quote.
 - **a.** In the left navigation pane of the Salesforce Setup page, scroll to the Build section, expand **Develop**, and then click **Platform Events**.
 - **b.** On the Platform Events page, click **New Platform Event**.
 - **c.** On the New Platform Event page, enter the following information:

Field	Information to Enter
Label	Enter the field label. For example, TransactionId_CPQ.
Plural	Enter the plural form of the field label.
Object Name	This field is automatically populated.
Description	If required, enter a description for the new platform event.
Publish Behaviour	 Select one of the following options: Publish After Commit The event message is published only after a transaction commits successfully. Subsequently, the subscriber receives the event message after the data is committed. If the transaction fails, the event message isn't published. Publish Immediately The event message is published when the publish call executes, irrespective of whether the transaction succeeds or not. Subsequently, the subscriber receives the event message before the data is committed.

- d. In the Deployment Status section, select **Deployed**.
- e. Click Save.

The Platform Event page displays the details of the event.

- f. Create a custom field.
 - In the Custom Fields & Relationships section of the Platform Event page, click New.



- ii. On the resultant New Custom Field page, select the **Text** radio button, and click **Next**.
- iii. Enter the following details:

Field	Information to Enter
Field Label	Enter a name for the field label. For example, QuoteID.
Length	Enter the maximum length that you want to allow for the text field. For example, 255.
Field Name	This field is automatically populated.
Description	If required, enter the description of the custom field.

- iv. Click Save.
- g. Create another custom field by repeating the steps a f, but enter a different label name, for example TransactionId CPQ.
- 8. Create and activate a process that triggers the platform event.
 - a. On the left navigation pane of the Salesforce Setup page, scroll to the Build section. Expand Create and then Workflow and Approvals, and click Process Builder.
 - b. Create a process.
 - i. On the Process Builder page, click New.
 - ii. Enter the following details in the New Process dialog:

Field	Information to Enter
Process Name	Enter a name for the process. For example, CPQ.
API Name	This field is automatically populated.
Description	If required, enter a suitable description for the process.
The process starts when	Select A record changes.

iii. Click Save.

The Process Builder page now displays a default process flow.

- iv. Click the Add Criteria decision box in the flowchart.
 The Process Builder displays Define Criteria for this Action Group section.
- v. In the Immediate Actions box beside the TRUE arrow, click Add Action. The Process Builder displays Select and Define Action section for adding actions.
- vi. Click Save.
- c. Configure the process trigger. After you have created the process in the previous step, you have to configure the rules required for the process to trigger the platform event.
 - i. Click the **Add Object** box in the process flowchart.
 - ii. From the Choose Object and Specify When to Start the Process pane that opens, select the following and click **Save**.



Field	Information to Enter
Object	Select Quote.
Start the process	Select When a record is created or edited.

- **d.** Add the process criteria. You can specify the criteria that must be true for the process to execute associated actions.
 - i. Click Add Criteria in the decision box.
 - ii. In the Define Criteria for this Action Group pane that opens, provide the following information.

Field	Information to Enter
Criteria Name	Enter a name for the criteria.
Criteria for Executing Actions	Select Conditions are met.

iii. Create a condition to restrict the user from triggering the process. In the **Set Conditions** table, configure the following:

Field	Information to Enter
Field	Search for LastModifiedBy.CommunityNickname and select it.
Operator	Select Does not equal.
Туре	Select String .
Value	Enter the nickname of the Salesforce user who creates or edits the quotes.

iv. Create another condition to ensure that the process gets triggered when line items are available.

Click **Add Row** and configure the following:

Field	Information to Enter
Field	Search for LineItemCount and select it.
Operator	Select Greater than.
Туре	Select Number.
Value	Enter 0.

- v. From the Conditions options list, select All of the conditions are met (AND).
- vi. Expand the Advanced drop-down list, and for Do you want to execute the actions only when specified changes are made to the record? select Yes.
- vii. Click Save.
- e. Add action to your process.
 - i. In the **Immediate Actions** box beside the **TRUE** arrow, click **Add Action**.
 - ii. In the Select and Define Action pane, from the Action Type drop-down list, select Create a Record.
 - iii. From the Record Type drop-down list, enter the name of the platform event you have created in step 7.In the Set Field Values section, create two rows with the following values:



For row 1, set the following values:

Field	Information to Enter
Field	Type the name of the custom field you have created in sub-step f of step 6. For example, QuoteID.
Туре	Select Field Reference.
Value	Click this field and on the resultant Select a Field dialog, select Quote ID from the drop-down, and click Choose .

For row 2, set the following values:

Field	Information to Enter
Field	Type the name of the custom field you have created in sub-step g of step 6. For example, TransactionId_CPQ.
Туре	Select Field Reference.
Value	Click this field and on the resultant Select a Field dialog, select TransactionId_CPQ from the dropdown, and click Choose .

iv. Click Save.

- f. Activate the process.
 - i. On the top right of the Process Builder page, click Activate.
 - ii. In the Activate Version dialog, click Confirm.

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle CPQ REST Connection
- Configure the Oracle CPQ SOAP Connection
- Configure the Oracle Salesforce Connection
- Configure the Lookup Table
- Configure the Integration



Configure the Oracle CPQ REST Connection

- 1. On the Configuration Editor page, select **Oracle CPQ Connection**.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
Connection Type	Select REST Catalog URL.
Connection URL	Enter the instance URL of Oracle CPQ. For example, https:// <oracleinstance>.com/rest/v8/metadata-catalog.</oracleinstance>
	See Configure Oracle CPQ.

4. 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 in Oracle CPQ.
	See Configure Oracle CPQ.
Password	Enter the password of the account created for Oracle Integration in Oracle CPQ. See Configure Oracle CPQ.

- 5. Click Save. If prompted, click Save for a second time.
- **6.** Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle CPQ SOAP Connection

- 1. On the Configuration Editor page, select **Oracle CPQ SOAP Connection**.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
Connection Type	Select SOAP WSDL URL.



Field	Information to Enter
Connection URL	Enter the URL of the Oracle CPQ instance. For example, https:// <oracleinstance>.com/rest/v8/metadata-catalog.</oracleinstance>
	See Configure Oracle CPQ.

4. 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 in Oracle CPQ.
	See Configure Oracle CPQ.
Password	Enter the password of the account created for Oracle Integration in Oracle CPQ.
	See Configure Oracle CPQ.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- **6.** Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle Salesforce Connection

- 1. On the Configuration Editor page, select Oracle Salesforce Connection.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection 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.
	See step 3 in Configure Salesforce.

4. 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 Configure Salesforce.



Field	Information to Enter
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.

- Click **Save**. If prompted, click **Save** for a second time.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click Test again.

A confirmation message is displayed if your test is successful.

7. Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Lookup Table

Configure the ORCL-BRT-SFDC_CPQ_CONSTANTS lookup table to replace the default Salesforce ID value with the specific Salesforce ID for the Standard Book Price of your Salesforce instance..

On the Configuration Editor page, select the lookup table and click **Edit**



The ORCL-BRT-SFDC_CPQ_CONSTANTS lookup table appears. The lookup table contains a mapping of constant values used internally by the system, and names that you can associate with these constants. Additionally, it contains a default value (Salesforce ID) that is mapped to the Standard Price Book of Salesforce.

- Replace the default Salesforce ID in the SFDC_ID column with your Salesforce instance's Standard Price Book Salesforce ID.
 - For more information on how to identify the Salesforce ID of your Salesforce instance, see step 5 in Configure Salesforce.
- Click **Save**. If prompted, click **Save** for a second time.
- 4. Click **Back** to return to the Configuration Editor page.

Configure the Integration

Configure the integration so that Oracle Integration susbcribes to Salesforce platform events that are published, and processes the appropriate platform event that you have created for this recipe.

- 1. On the Configuration Editor page, select Oracle Salesforce CPQ Quote2Transaction Sync integration.
- 2. Click Edit



Click the Salesforce trigger (GetQuote_PE), and then click Edit .
 The Configure Salesforce.com Endpoint wizard appears.

- 4. Click Next.
- 5. On the Salesforce Trigger Action page, select **Streaming API**, and click **Next**.
- 6. On the Streaming API Configuration page, select **Subscribe to platform events** published on **Salesforce.com**.
- 7. In the **Select a platform event** text box, select the platform event that you created earlier.

See step 7 in Configure Salesforce.

Click Next.

The Salesforce.com Endpoint Configuration Summary dialog appears.

- 9. Click Done.
- Click Save. If prompted, click Save for a second time.
 Oracle Integration displays a message indicating that the save operation is successful.
- 11. Click Close.

Activate and Run the Recipe

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

Note that before testing the integration flows, ensure that you've an opportunity created in your Oracle CPQ instance as well as in your Salesforce instance. Transactions in Oracle CPQ and quotes in Salesforce can only be created for an opportunity.

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- Test the Oracle CPQ to Salesforce integration flow.
 - a. Create a transaction in Oracle CPQ Cloud.
 - i. Log in to the Oracle CPQ Cloud instance.
 - ii. Click the user icon on the top right of the page, and select the opportunity where you want to create the new transaction.
 - iii. In the resulting page, click New Transaction.
 - iv. In the Line Iten Details tab, click Quick Add.
 - v. In the Quick Add dialog, search for an item with the product code.
 - vi. Select the product and click **OK**.
 - vii. Finally, click Submit_to_SFDC.
 - **b.** Monitor the execution of the integration flow in Oracle Integration.
 - i. On the Configuration Editor page, select the integration.
 - ii. Click Run , and then click Track Instances.



On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully. The recipe fetches the new transaction related information from Oracle CPQ Cloud and triggers the creation of a quote in Salesforce.

- c. Check if a corresponding quote is created in Salesforce.
 - i. Log in to your Salesforce instance.



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

ii. On the Salesforce Setup page, click the **Opportunities** tab.



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

- iii. Check if the new quote was created.
- 3. Test the Salesforce to Oracle CPQ Cloud integration flow.
 - a. Create a quote with line-items in Salesforce.
 - i. Log in to your Salesforce instance.

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.

- ii. Create a new quote for an opportunity.
 - Open the opportunity.
 - Scroll to the Quotes section and click New Quote.
 - In the **Quote Name** field, enter a name for the quote.
 - Click Save.
- iii. Add a line item for the quote.
 - On the Quote details page, scroll to the Quote Line Items section and click Add Line Item.
 - In the Choose Price Book dialog, select Standard Price Book from the Price Book drop-down list, and click Save.
- iv. On the Product Selection for Quote page, select the required product and click Select.



- v. Enter the quantity in the Quantity field.
- vi. Click Save.
- **b.** Monitor the execution of the integration flow in Oracle Integration.
 - i. On the Configuration Editor page, select the integration flow.
 - ii. Click Run , then click Track Instances.
 On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully. The recipe fetches all the quote related information from Salesforce and triggers the creation of a transaction in Oracle CPQ Cloud.
- c. Check if the corresponding transaction is created in Oracle CPQ Cloud.
 - i. Log in to your Oracle CPQ instance.
 - ii. Click the user icon on the top right of the page, and select the opportunity.
 - iii. Check if the new transaction was created.

Related Documentation

- Using the Oracle CPQ Adapter with Oracle Integration
- Using the Salesforce Adapter with Oracle Integration

Sync Incidents and Resources Between Oracle Field Service Cloud and Oracle B2C Service

Use this recipe to synchronize incidents and resources between Oracle Field Service Cloud and Oracle B2C Service.



This recipe is available as **Oracle Field Service Cloud** — **Oracle B2C Service | Sync Incidents and Resources** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe supports bi-directional, data synchronization of incidents and resources between Oracle Field Service Cloud and Oracle B2C Service. It includes out-of-the-box data mapping of salient attributes.

Oracle Field Service Cloud manages the most intimate customer service channel: field service appointments. These interactions may be the only time your customer engages with your organization face-to-face, and thus have a significant impact on customer relationships.

Oracle B2C Service helps you serve your customers at their point of need, whether that customer is searching for assistance via your corporate social channels, an email inquiry, a voice call to a support agent, or a website chat.

This recipe includes the following features:



- End-to-end integration between Oracle Field Service Cloud and Oracle B2C Service as an out-of-the-box solution using Oracle Integration
- Automatic creation of activities in Oracle Field Service Cloud when an incident is created in Oracle B2C Service
- Synchronize modified fields between incidents and activities
- Create and update resources in Oracle B2C Service when created or updated in Oracle Field Service Cloud

System and Access Requirements

- Oracle Field Service Cloud, 17.2 or later
- Oracle B2C Service, 16.8 or later
- Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

Oracle Field Service Cloud / Oracle Service Cloud Integration using ICS

Sync Orgs and Contacts Between Oracle B2C Service and Oracle Marketing Cloud

Use this recipe to synchronize the organization and contact information between Oracle B2C Service and Oracle Marketing Cloud.



This recipe is available as **Oracle B2C Service** — **Oracle Marketing Cloud** | **Sync Orgs and Contacts**

in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

Oracle B2C Service makes it possible for you to better understand your customers and quickly deliver the right answers at the right time through preferred service channels and devices. Oracle Marketing Cloud provides lead and campaign management tools that help you engage the right audience at the right time in the buyer's journey.

This recipe synchronizes the organization and contact information between the Oracle service and marketing applications. The integration provides the following features:

 Create or update contacts in Oracle B2C Service based on contact events in Oracle CX Marketing



- Create or update contacts in Oracle Marketing Cloud based on contact creation or update in Oracle B2C Service
- Create or update accounts in Oracle Marketing Cloud based on organization creation or update in Oracle B2C Service

System and Access Requirements

- Oracle B2C Service, 16.11 or later
- Oracle Marketing Cloud
- Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

Integrating Oracle Marketing Cloud (Eloqua) with Oracle Service Cloud (RightNow)

Enterprise Messaging Recipes

These recipes help you get started with messaging applications.

Topics:

Publish and Subscribe to the Oracle Cloud Infrastructure Streaming Service

Publish and Subscribe to the Oracle Cloud Infrastructure Streaming Service

Use these recipes to send messages from Oracle Integration to the Oracle Cloud Infrastructure Streaming Service and receive messages from the Oracle Cloud Infrastructure Streaming Service.



This recipe is available as **Oracle Cloud Infrastructure Streaming - Publish and Subscribe** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

Recipe Name Description **Configuration Required Oracle Cloud** This integration with Oracle To use the recipe: Infrastructure Cloud Infrastructure Ensure you have met the prerequisites. See Prerequisites for Streaming -Streaming Service is Creating a Connection. Publish triggered by a schedule and publishes messages Configure the connection parameters to point to your Oracle from Oracle Integration to Cloud Infrastructure Streaming Service. For detailed the specified stream pool. information on how to configure the parameters, see Create a Connection. The connection for this **Bootstrap Server** integration is the same as that for the recipe Oracle Security Policy: SASL PLAIN Over SSL OSS Cloud Infrastructure SASL User Name Streaming - Subscribe. SASL Password TrustStore Agent Group, if the service is hosted on your premises behind the fire wall 3. In the integration, edit the schedule definition to specify how often messages are retrieved from Oracle Cloud Infrastructure Streaming Service. For detailed information on how to edit a schedule definition, see Define the Integration Schedule. Edit the integration and edit the **Publish** endpoint. Specify the stream and partition to which messages are to be sent and be sure to include a sample of the message structure. The integration has the following sample message structure that you'll need to replace. For details, see Stream and Partition Page and Message Structure Page. "Id" : "123", [ns2:getFlowId ()] "Country": "US", "State": "MA", "County": "Worcester" Activate the integration. For instructions, see Activate an Integration.

Monitor the integration. You can track the integration and monitor the sent messages in the activity stream. For

instructions, see View the Activity Stream.



Recipe Name
Oracle Cloud Infrastructure Streaming - Subscribe

Description

This integration with Oracle To use the recipe: Cloud Infrastructure Streaming Service is triggered by a schedule and consumes messages from the specified stream pool.

The connection for this integration is the same as that for the recipe Oracle Cloud Infrastructure Streaming - Publish.

Configuration Required

- Ensure you have met the prerequisites. See Prerequisites for Creating a Connection.
- Configure the connection parameters to point to your Oracle Cloud Infrastructure Streaming Service. For detailed information on how to configure the parameters, see Create a Connection.
 - **Bootstrap Server**
 - Security Policy: SASL PLAIN Over SSL OSS
 - SASL User Name
 - SASL Password
 - TrustStore
 - Agent Group, if the service is hosted on your premises behind the fire wall
- Edit the schedule definition in the integration to specify how often messages are retrieved from Oracle Cloud Infrastructure Streaming Service. For detailed information on how to edit a schedule definition, see Define the Integration Schedule.
- Activate the integration. For instructions, see Activate an Integration.
- Monitor the integration. You can track the integration and monitor the consumed messages from the activity stream. For instructions, see View the Activity Stream. Here's an example of what the activity stream looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<ConsumeResponse xmlns="http://xmlns.oracle.com/cloud/</pre>
adapter/ocistreaming/ConsumeFrom REQUEST/types">
  <MessageOutCollection xmlns="http://xmlns.oracle.com/</pre>
cloud/adapter/kafka/types" xmlns:NS1="http://
xmlns.oracle.com/cloud/adapter/kafka/types"
NS1:status="SUCCESS">
     <Message>
    <headers/>
    <request-wrapper>
      <content xmlns="http://xmlns.oracle.com/cloud/</pre>
adapter/ocistreaming/ConsumeFrom/types">
         <Id>7.2803233E7</Id>
           <Country>US</Country>
         <State>MA</State>
         <county>Worcester</county>
      </content>
    </request-wrapper>
    <config-parameters>
      <partition>0</partition>
      <offset>25</offset>
    </config-parameters>
    <error-details/>
    <status>SUCCESS</status>
    </Message>
  </MessageOutCollection>
</ConsumeResponse>
```



System and Access Requirements

- Oracle Integration
- Oracle Cloud Infrastructure Streaming Service

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

- Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration
- Overview of Streaming

Enterprise Resource Planning (ERP) Recipes

Review these enterprise resource planning (ERP) recipes if you're looking to integrate applications that manage your day-to-day business activities such as transferring files, importing financials, and managing the supply chain (inventory, purchase orders, receiving, and shipping).

Topics:

- Automate File Transfer an FTP server to Oracle EPM Cloud
- Create Customer Records in Oracle NetSuite
- Export Purchase Orders from Oracle ERP Cloud to an FTP Server
- Extract Bulk Data from Oracle ERP Cloud to an FTP server
- Import Files from an FTP Server to Oracle NetSuite
- Import Financial Journal Entries from Amazon S3 to Oracle ERP Cloud
- · Import Financial Journal Entries from an FTP server to Oracle ERP Cloud
- Process Inventory, Order, and Shipping Info Between Oracle ERP/SCM Cloud and Oracle WMS Cloud
- Send Order Fulfillment Notifications from Oracle ERP Cloud Using Microsoft Office 365 Outlook
- Synchronize Inventory Items Between Oracle Product Hub Cloud and Oracle E-Business Suite
- Synchronize Opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite Orders
- Synchronize Supplier Updates Between Oracle Procurement Cloud and Oracle ATP



Automate File Transfer an FTP server to Oracle EPM Cloud

Use this recipe to automate file transfer from an FTP server to Oracle Enterprise Performance Management Cloud (Oracle EPM Cloud).



This recipe is available as **Oracle EPM Cloud** — **FTP Server | Automate File Transfer** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe demonstrates how easily you can transfer a file from any FTP/sFTP location to Oracle EPM Cloud using Oracle Integration.

The integration uses a scheduled orchestration pattern to automatically transfer the file on a regular basis. It uses the standard FTP and REST adapters available in Oracle Integration to read the file from the source location (FTP server) and then transfer it to the destination (Oracle EPM Cloud).

You store the file to be transferred 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 Generation 2*.

System and Access Requirements

- Oracle Integration, 18.1.5 or later
- Oracle EPM Cloud, 18.03 or later
- A secure FTP (sFTP) server or File Server.

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

- File Transfer EPM-FTP Using Integration Cloud Sample OIC Integration Flow (File transfer to EPM Cloud from FTP location) Implementation Guide
- Using File Server in Oracle Integration Generation 2



Create Customer Records in Oracle NetSuite

Use this recipe to create a customer record in Oracle NetSuite through a REST request from an external application.



This recipe is available as **REST** — **Oracle NetSuite | Create Customer Records** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be errorfree. No support is provided for this recipe.

Overview

This recipe creates a new customer record in Oracle NetSuite when it receives a REST request with the customer data from an external application or from another integration flow in Oracle Integration. It contains an app-driven orchestration integration, which uses the standard REST Adapter and the Oracle NetSuite Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate the integration in the package and send a REST request with the customer data to the integration's endpoint URL from an external application. When triggered, the integration flow creates the customer record in Oracle NetSuite using the data it receives and returns the NetSuite internal ID of the newly-created record as a response.



You can create only one customer record at a time with this recipe.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39447) or higher
- Oracle NetSuite, Version 2018.1.0
- An account on Oracle NetSuite with the Administrator role

Topics:

- Before You Install the Recipe
- · Install and Configure the Recipe
- · Activate and Run the Recipe

Before You Install the Recipe

To successfully connect to NetSuite using Oracle Integration and create customer records, you must perform certain general configurations on your NetSuite instance



and some configurations specific to the security policy you'll use to connect. In this recipe, you'll use the token-based authentication (TBA) security policy.

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

- Perform the general configurations necessary to connect to 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	
Lists	Currency	Full	
Lists	Customers	Full	
Lists	Subsidiaries	Full	
Lists	Contacts	Full	
Setup	SOAP Web Services	Full	
Setup	User Access Tokens	Full	
Setup	Log in using Access Tokens	Full	

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST NetSuite Connection
- Configure the Oracle NetSuite Connection



Note:

Optionally, you can configure the following lookup table in the package.

ORCL-BRT-EC_NS_SETTINGS: This lookup table contains a
 sample mapping of country codes. Map the country codes of your
 application to the NetSuite country codes. Additionally, this lookup
 table contains a default value for the internal ID of the NetSuite
 subsidiary to which the new customer records will belong. To
 associate the new customer records with the correct subsidiary on
 your NetSuite instance, enter the internal ID of the preferred
 subsidiary against the subsidiary field in the table. Also, against the
 ToEmailID field, you can enter an email ID on which you'd like to
 receive notifications related to the integration's execution.

Configure the Oracle REST NetSuite Connection

Test and save the Oracle REST NetSuite connection.

- 1. On the Configuration Editor page, select Oracle REST NetSuite Connection.
- 2. Click Edit
- On the Oracle REST NetSuite Connection page, click Test to ensure that your connection is correctly configured.

A confirmation message is displayed if your test is successful.

- 4. Click Save. If prompted, click Save for a second time.
- Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Oracle NetSuite Connection

- 1. On the Configuration Editor page, select **Oracle NetSuite Connection**.
- 2. Click Edit .

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
WSDL URL	<pre>Enter the NetSuite WSDL URL; for example, https://webservices.netsuite.com/wsdl/ <netsuite_application_version>/ netsuite.wsdl.</netsuite_application_version></pre>
	where <netsuite_application_version> is the version of the NetSuite application. For example, v2015_1_0. See Assemble the Oracle NetSuite WSDL URL.</netsuite_application_version>

4. 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.
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.
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.



You must enter this information in capital

- 5. Click Save. If prompted, click Save for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Activate and Run the Recipe

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

1. On the Configuration Editor page, click Activate in the title bar. In the Activate Package dialog, click Activate again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- Run the recipe from an external application.
 - a. On the Configuration Editor page, select the integration and click **Run** .



- b. On the resulting pop-up dialog, click the **Metadata URL** link.
- c. From the new tab 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 customer data. Provide the customer data in the POST request's Body field. See the following step for sample customer data.

The integration creates the customer record in Oracle NetSuite and returns the internal ID of the newly-created record as a response.

- Test the recipe in Oracle Integration.
 - a. On the Configuration Editor page, select the integration.
 - b. Click Run , then click Test.



The page to test the integration with sample data is displayed.

c. In the Request section, click **Body** and enter sample customer data. For example:

```
{
  "zip": "5",
  "country": "Ireland",
  "address2": "addr2",
  "city": "Dublin",
  "address1": "addr1",
  "last_name": "London",
  "company": " Acme Corporation",
  "first_name": "James",
  "email": "J.London@acmecorp.com"
}
```

d. Click Test.

The integration now creates the customer record in Oracle NetSuite. The Activity Stream pane appears displaying the status of the integration instance's execution.

- e. In the Response section of the test page, under the **Body** tab, you'll find the internal ID of the newly-created customer record in NetSuite.
- Log in to your NetSuite instance as an Administrator and check for the new customer record.
 - a. On the NetSuite home page, select Lists, then Relationships, and then Customers.
 - **b.** On the Customers page, select **Recently Created** in the **QUICK SORT** field to list the recently-created customer records first.
 - Use the internal ID value you obtained as a response to verify if the required customer record has been created.

Related Documentation

- Using the Oracle NetSuite Adapter with Oracle Integration
- Using the REST Adapter with Oracle Integration

Export Purchase Orders from Oracle ERP Cloud to an FTP Server

Use this recipe to subscribe to an Oracle ERP Cloud business event. This recipe subscribes to a business event that Oracle ERP Cloud raises when a purchase order is created, and writes the purchase order data to a file on an FTP file server.



This recipe is available as **ERP Cloud - FTP Server | Business Event** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

This recipe registers with Oracle ERP Cloud Business Event services. When a purchase order is created, Oracle ERP Cloud Business Event services invokes the integration that in turn, adds the purchase order data to a file on an FTP file server.

You can store the file in File Server, an SFTP server embedded in Oracle Integration.

A purchase event in Oracle ERP Cloud triggers the integration. The integration reads from the lookup values where to store the CSV file in the FTP file server, then maps the values of the purchase order to the fields in the CSV file, and finally appends a line with the purchase order values to the file. If the file doesn't exist, it creates a new file.

The CSV file contains the following fields:

- Creation Date
- Order Number
- Supplier
- Buyer Name
- Currency
- Total Amount

The Order Number field is mandatory. The rest of the fields are optional.

If there is an issue while running the integration, the fault handler adds the issue and its cause to the logs. You can use these logs to troubleshoot the issue.

The integration that results from installing this recipe is named: Oracle ERP Cloud FTP Server Business Event.

System and Access Requirements

- Oracle Integration
- Oracle ERP Cloud
- File Server

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

After you install the recipe, and before you run the integration you must:

- Configure Oracle ERP Cloud and Oracle Integration
- Enable File Server (only if you are using File Server)
- Configure the Integration Lookups
- Configure the ERP connection
- Configure the FTP connection

Configure Oracle ERP Cloud and Oracle Integration

You must subscribe to Oracle ERP Cloud business events:



- 1. Create a Fusion Applications user to use in the Oracle ERP Cloud connection, and assign it the following roles:
 - Integration Specialist
 - AttachmentsUser
 - FND MANAGE CATALOG SERVICE PRIV

Notice that you can only add this role to a custom role. For more information about creating custom roles, see Custom Roles in Securing ERP.

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

For more information about these roles, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration*.

- 2. Create a CSF (Credential Set Framework) key for Oracle ERP Cloud to use when calling the integration:
 - Use an Oracle Integration user with the ServiceInvoker role to create the CSF key.
 - Follow steps 1 to 5 in Configure Oracle ERP Cloud for Event Subscriptions.

The purchase order event is enabled by default in Oracle ERP Cloud.

For information on how to enable other events, see Supported SCM and Procurement Business Events in *Using the Oracle ERP Cloud Adapter with Oracle Integration*.

Enable File Server

If you are using an external FTP file server to store the CSV file, skip this section. Just make sure you have access to your FTP file server.

If you are using File Server, you must enable it for the user that you'll use to access File Server.

Before following these steps, make sure File Server is enabled. For information on how to enable File Server, see Enable File Server in *Using File Server in Oracle Integration Generation 2*.

- From the Main menu, select Integrations, then Settings, then File Server, and then Users.
- 2. Search for the user that you'll use to access File Server.

You can use the same user that you used to install the recipe.

- 3. Make sure the switch in the **Enable** column is on.
- **4.** Make a note of the **Home Folder** value for this user. You will need this value to you configure the directory.

Configure the Integration Lookups

You must edit the value for the directory lookup to use your home folder:

- 1. From the Main menu, select Integrations, then Lookups.
- 2. Open ORCL-BRT-FTP_FOLDER.
- 3. In the **Property_Value** column for the **directory** property name, replace / InputDirectory with the path to the folder where you want to store the CSV file.



- If you are using File Server, enter the path that you wrote down when you enabled File Server for your user.
- 4. In the **Property_Value** column for the **filename** property name, replace CVS with the name you want to use for the file that stores the purchase order data. For example: PurchaseOrders.csv

Configure the ERP Connection

You must configure the ERP connection to use the instance and credentials for your ERP instance:

- 1. From the Main menu, select Integrations, then Connections.
- 2. Locate and open Oracle ERP Cloud Connection.
- 3. In the **Connection Properties** section, in the **ERP Cloud Host** field, enter the URL to access your Oracle ERP Cloud instance.
- 4. In the **Security** section, select a security policy and enter the credentials to access Oracle ERP Cloud.
 - For more information, see Configure Connection Security in *Using the Oracle ERP Cloud Adapter with Oracle Integration*.
- 5. Click **Save**, and then click < to go back to the **Connections** list.

Configure the FTP Connection

You must configure the FTP connection to use your server and credentials:

- 1. If you are using an external FTP, find out the IP and port where the FTP server runs.
 - If you are using File Server, find its IP and port:
 - a. From the Main menu, select Integrations, then Settings, then File Server, and then Settings.
 - **b.** From the **General** section, copy the IP and port.
- 2. From the Main menu, select Integrations, then Connections.
- 3. Locate and open Oracle FTP Connection.
- 4. In the Connection Properties section enter the values for the following fields:
 - **a. FTP Server Host Address**: the IP for your external FTP file server or the IP you copied from the File Server settings.
 - **b. FTP Server Port**: the port for your external FTP file server or the port you copied from the File Server settings.
 - c. SFTP Connection: Yes
- In the Security section, enter the username and password for your Oracle Integration instance.

Test the Recipe

After configuring the recipe, you can activate it and test it:

- 1. Activate the integration.
 - For information on how to activate an integration, see Activate an Integration in *Using Integrations in Oracle Integration*.



- 2. Create a purchase order in Oracle ERP Cloud.
- 3. Review the **Monitoring** page to verify the integration ran successfully:
 - a. In Oracle Integration, from the **Main** menu, select **Monitoring**, then **Integrations**, and then **Tracking**.
 - **b.** Search for your integration.
 - **c.** When the integration is complete, click the primary identifier to open the integration instance.

The flow is highlighted in green to indicate that it ran successfully.

- d. Click and select View Activity Stream.
- e. Review the flow and messages for your integration.
- f. Click Close when you finish reviewing.
- 4. Connect to the file server using an FTP client, open the directory where you stored the CSV file, and then open the CVS and verify that the purchase order data was added to the file.

You can connect to File Server using an FTP client with the information you wrote down in Enable File Server.

For information on how to configure folder access in File Server, see Configure Folders and View List of Files in *Using File Server in Oracle Integration Generation* 2.

Related Documentation

To learn more about the Oracle ERP Cloud adapter, the Oracle FTP adapter, and File Server see:

- Using the Oracle ERP Cloud Adapter with Oracle Integration
- Using the FTP Adapter with Oracle Integration
- Using File Server in Oracle Integration Generation 2

Extract Bulk Data from Oracle ERP Cloud to an FTP server

Use this recipe to export bulk data from Oracle Enterprise Resource Planning (ERP) Cloud and import it into an FTP server.



This recipe is available as **Oracle ERP Cloud** — **FTP Server | Extract Bulk Data** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe extracts BI Publisher (BIP) report data from Oracle ERP Cloud in an asynchronous manner. The integration uses a scheduled orchestration pattern so that you can schedule when to extract the data extract and how often. The extracted



reports are then transferred and uploaded to an FTP server. The reports are now available to users and applications directly from the FTP location. If required, you can enrich or edit the extracted report data within the integration.

The sample integration performs these actions:

- Query bulk data from Oracle ERP Cloud.
- Receive the data via the ERP trigger.
- Invoke it through the SOAP Adapter and get the data from a business event.
- Complete the process. If the import completes successfully as per the Oracle ERP Cloud callback, then upload the file received in the callback to the FTP server. If the import fails, then send an email notification.

You can upload the extracted reports to 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 Generation 2*.

System and Access Requirements

- Oracle Integration, Version 19.2.3.0.0 (190518.1400.28490)
- Oracle ERP Cloud, Version 19A (11.13.19.01.0)
- A secure FTP (sFTP) server or File Server

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

- Overview | Extract Bulk Data from ERP Cloud_Async
- Use Cases | Extract Bulk Data from ERP Cloud Async
- User Guide | Extract Bulk Data from ERP Cloud_Async
- Using File Server in Oracle Integration Generation 2

Import Files from an FTP Server to Oracle NetSuite

Use this recipe to import files from an FTP server to the NetSuite File Cabinet.



This recipe is available as **FTP Server** — **Oracle NetSuite | Import Files** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

This recipe loads files from an FTP server to the NetSuite File Cabinet according to a schedule specified in Oracle Integration. It uses the standard FTP Adapter and the Oracle NetSuite Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate and run the integration flow of the package manually or specify an execution schedule for it. When triggered, the integration flow reads the files from the FTP server and copies them to your NetSuite File Cabinet. After successfully copying the files, the integration flow deletes the files from the FTP server to avoid copying the same files in each scheduled execution.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462) or higher
- Oracle NetSuite, Version 2019.1.0 or higher
- An account on Oracle NetSuite with the Administrator role
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite instance and FTP server in order to successfully connect to these external systems using Oracle Integration and transfer files between them.

- 1. Access Your FTP Server and Upload the Files
- 2. Configure NetSuite

Access Your FTP Server and Upload the Files

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

- Log in to the server using your user name and password through an FTP client; for example, FileZilla.
- 2. Create a directory (ftp-ns-transfer) on the server from which to read the files. Note down the path of this directory.
- 3. Upload a sample file (test-file.pdf) into the directory you've created. Note down the format of the file.



Configure NetSuite

To access NetSuite using Oracle Integration and copy files, you must perform certain configuration tasks on your NetSuite instance.

To successfully connect to NetSuite, you must perform certain general configurations on your NetSuite instance and some configurations specific to the security policy you'll use to connect. In this recipe, you'll use the token-based authentication (TBA) security policy.

In addition, you must create a directory in the NetSuite File Cabinet into which you'll copy the files from the FTP server.

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

- 1. Perform the general configurations necessary to connect to 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
Lists	Documents and Files	Full
Setup	SOAP Web Services	Full
Setup	User Access Tokens	Full

- 3. Create a directory in the NetSuite File Cabinet to hold the imported files.
 - a. On the NetSuite home page, select **Documents**, then **Files**, and then **File Cabinet**.
 - b. On the File Cabinet page, click **New Folder**.
 - c. On the Document Folder page:
 - i. Enter a name for the folder, for example, ftp-ns.
 - ii. Click Save.

The folder is created, and the folder's page is displayed.

- d. Return to the File Cabinet page. Select **Documents**, then **Files**, and then **File** Cabinet from the ribbon.
- e. Note down the INTERNAL ID value displayed next to your directory. If the INTERNAL ID column is not displayed in your File Cabinet, you must enable it. See Setting the Internal ID Preference in Oracle NetSuite.

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the **Accelerators & Recipes** section.
- Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.



5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle FTP Read-Write Connection
- Configure the Oracle NetSuite Connection
- Configure the Integration

Configure the Oracle FTP Read-Write Connection

- 1. On the Configuration Editor page, select **Oracle FTP Read Write Connection**.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection 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.

4. 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.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- Click Test to ensure that your connection is successfully configured. In the resulting dialog, click Test again.

A confirmation message is displayed if your test is successful.

 Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Oracle NetSuite Connection

- 1. On the Configuration Editor page, select Oracle NetSuite Connection.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.



Field	Information to Enter
WSDL URL	Enter the NetSuite WSDL URL; for example, https://webservices.netsuite.com/wsdl/ <netsuite_application_version>/netsuite.wsdl.</netsuite_application_version>
	where <netsuite_application_version> is the version of the NetSuite application. For example, v2015_1_0. See Assemble the Oracle NetSuite WSDL URL.</netsuite_application_version>

4. 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.
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.
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.



You must enter this information in capital letters.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Integration

Create a schedule for your integration to pick up files from the FTP server at a date, time, and frequency of your choosing.

- 1. On the Configuration Editor page, select the integration.
- Click Actions , then click Add Schedule.
 The page for defining the execution schedule is displayed.
- 3. For this example, let's create a basic schedule, so leave the **Simple** radio button selected under the **Define recurrence** section.
- 4. In the Frequency field, select Hours and Minutes, and change the hours field to 6.



- You've now created a schedule that runs the integration every six hours. Similarly, you can schedule the integration to run on particular days or weeks.
- 5. Additionally, you can specify the start and end dates for this schedule. In the This schedule is effective section, click the From field and select Modify start date to specify a date and time from when this schedule should run. Similarly, specify an expiry date and time in the Until field. In addition, you can also specify your preferred timezone.
- 6. Click **Save**, and then click **Back** twice to return to the Configuration Editor page. To create advanced schedules, see Define the Integration Schedule.

Activate and Run the Recipe

After you've configured the connections and the integration flow, activate the recipe and run it.

- 1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.
 - A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.
- 2. Update the integration property values.
 - Configure the property values necessary for the successful execution of the integration.
 - Using the property values, you can specify the FTP server directory from which the files must be read, the format of the files to be read, and the Oracle NetSuite directory into which the files must be written.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Actions , then click Update Property Values.
 - **c.** In the Update Property Values pane, enter the appropriate values for the following integration properties.

Click on a property row to expand it, and enter a suitable value in the **New Value** field. Repeat for all property rows.

- FTPDirectory: Specify the path of the directory on your FTP server from where the files must be read. For example: /home/user2/FTP. See Access Your FTP Server and Upload the Files.
- FolderInternalId: Specify the internal ID of the Oracle NetSuite directory into which the files must be written. For example: 705. See Configure NetSuite.
- **FileNamePattern:** Enter the format of the files to be read from the FTP server. For example: *.pdf or *.* and so on.
- **NotificationMail (Optional):** Enter an email ID on which you'd like to be notified about the errors in the integration's execution.
- d. Click Submit.

A confirmation message is displayed informing that the integration properties have been updated successfully.

3. Run the recipe.



The recipe is now ready to use. Run it to import your files from the FTP server to the NetSuite instance.

- a. On the Configuration Editor page, select the integration flow.
- b. Click Run , then click Submit Now.
- c. In the resulting dialog, click Submit Now again. Note that the Ad hoc request radio button is selected by default.

You've now successfully submitted the integration for execution.

Log in to your NetSuite instance and check for the imported file(s) in your File Cabinet folder.

Related Documentation

- Using the FTP Adapter with Oracle Integration
- Using the Oracle NetSuite Adapter with Oracle Integration

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. The recipe uses File-Based Data Integration (FBDI).



This recipe is available as **Oracle ERP Cloud** — **Amazon S3 | Import Financial Journal Entries** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

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 message flow of business data goes from Amazon S3 through Oracle Integration to Oracle ERP Cloud, and then back to the Amazon S3 location.

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. In a scheduled orchestration, you can configure when and how often the download occurs.

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.

The integration uses the standard REST and Oracle ERP Cloud Adapters available in Oracle Integration.

System and Access Requirements

- Oracle Integration, Version 19.3.1.0.0 (190624.1100.29532) or later
- Oracle Fusion Cloud Enterprise Resource Planning, Version 19A (11.13.19.01.0) or later

Amazon S3

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

- Overview | V0 Import Journal Entries | Amazon S3/Oracle Financials Cloud (ERP Cloud)
- Use Case | V0 Import Journal Entries | Amazon S3/Oracle Financials Cloud (ERP Cloud)
- User Guide | V0 Import Journal Entries | Amazon S3/Oracle Financials Cloud (ERP Cloud)

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. The recipe uses File-Based Data Integration (FBDI).



This recipe is available as **Oracle ERP Cloud** — **FTP Server | Import Financial Journal Entries** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe reads a file from an FTP location, does a simple transformation, and then imports the financial journal entries into Oracle ERP Cloud. The integration flow can be invoked explicitly either as a web service or as a scheduled orchestration.

The sample integration performs the following actions:

- Read the ledger-entries file from the FTP location.
- Unzip the file.
- Stage the file in Oracle Integration and read each entry.
- Transform the data as required. For example, this sample integration changes the value in the currency column.
- Use the Oracle ERP Cloud Adapter to send the file to Oracle ERP Cloud, configure a callback, and invoke the import process with the required parameters.
- Report the results. After the import is completed, 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 in *Using File Server in Oracle Integration Generation 2*.

System and Access Requirements

- Oracle Integration
- Oracle ERP Cloud
- A secure FTP (sFTP) server or File Server

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

- Overview | V0 Import Journal Entries | Cloud FTP Location/Oracle Financials Cloud (ERP Cloud)
- Use Case | V0 Import Journal Entries | Cloud FTP Location/Oracle Financials Cloud (ERP Cloud)
- User Guide | V0 Import Journal Entries | Cloud FTP Location/Oracle Financials Cloud (ERP Cloud)
- Using File Server in Oracle Integration Generation 2

Process Inventory, Order, and Shipping Info Between Oracle ERP/SCM Cloud and Oracle WMS Cloud

Use this recipe to process information related to inventory, purchase orders, receiving, and shipping. It demonstrates a sample integration between Oracle Enterprise Resource Planning (ERP)/Supply Chain & Manufacturing (SCM) Cloud and Oracle Warehouse Management (WMS) Cloud.



This recipe is available as **Oracle ERP/SCM Cloud** — **Oracle WMS Cloud** | **Process Inventory, Order, and Shipping Info** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe syncs inventory, and maps receipts to purchase orders, receipts to receiving, and shipments to orders.

It includes the following individual integrations in one package:

• OCWMS_INVENTORY_ADJUSTMENT: Makes inventory adjustments from Oracle WMS Cloud to Fusion inventory.



- OCWMS_RECEIPT_ADVICE: Takes purchase order receipts in Oracle Inventory Management Cloud and maps them to purchase orders in Oracle WMS Cloud.
- OCWMS_RECEIPT_CONFIRMATION: Maps receipt confirmation from Oracle WMS Cloud to Fusion receiving.
- OCWMS_SHIPMENT_REQUEST: Maps shipment requests from Fusion shipping to orders in Oracle WMS Cloud.
- OCWMS_SHIPMENT_CONFIRMATION: Maps shipments from Oracle WMS Cloud to the confirmation in Fusion shipping.
- OCWMS_ECHO_LGFDATA: Used for internal XML transformation. No user configuration is necessary.

The integration uses the standard REST Adapter available in Oracle Integration to create a connection to a specific instance of Oracle WMS Cloud. The connection is then used to create an integration, which calls Oracle WMS Cloud web services, as well as exposes web services that Oracle WMS Cloud can call to push out data. The sample flow uses the Oracle ERP Cloud Adapter to connect to ERP Cloud instances (Fusion/Cloud Inventory).

System and Access Requirements

- Oracle WMS Cloud, Version 9.0.0 or later
- Oracle Inventory Management Cloud, Version 18A or later
- Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

- Oracle Warehouse Management Cloud Integrating with Integration Cloud Services
- Using the Oracle ERP Cloud Adapter with Oracle Integration
- Using the REST Adapter with Oracle Integration

Send Order Fulfillment Notifications from Oracle ERP Cloud Using Microsoft Office 365 Outlook

Use this recipe to send email notifications when a sales order is fulfilled in Oracle ERP Cloud.



This recipe is available as **Oracle ERP Cloud** — **Microsoft Office 365 Outlook | Send Order Fulfillment Notifications** in the Integration Store.

Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

This recipe sends email notifications to order contacts when sales orders are fulfilled in Oracle ERP Cloud. The recipe uses the standard Oracle ERP Cloud Adapter and the standard Microsoft Office 365 Outlook Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate the integration flow of the package. When a sales order is fulfilled in Oracle ERP Cloud, it triggers the integration flow. The integration fetches the necessary details of the sales order and sends a notification email with the order fulfillment details. By default, it sends the notification email to the email address specified in the *toEmail* integration property. In addition, if the buyer's email address is present in the order, and the *DoNotEmailFlag* is set to "false", the notification email is also sent to the buyer.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462) or higher
- Oracle ERP Cloud
- Accounts on Oracle ERP Cloud with the Administrator role and the Sales user role (in order to create and fulfill orders)
- Access to Order Management in Oracle ERP Cloud
- Microsoft Azure
- An account on Microsoft Azure with the Administrator role

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance and Microsoft Email instance in order to successfully connect to these external systems using Oracle Integration and send order fulfillment notifications.

- 1. Configure Oracle ERP Cloud
- 2. Configure Microsoft Email

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.

 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, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Microsoft Email

Complete certain configuration tasks in your Microsoft Email instance to successfully connect to it from Oracle Integration.

- 1. Log in to https://azure.microsoft.com/ with administrator access.
- 2. In the upper right, click the user icon and select **Azure Portal**.
- 3. In the search field of the resulting page, enter app registration, and then select App registrations from the search results.

Alternatively, scroll to the Azure services section in the page, and click **App** registrations.

- 4. In the App registrations page, click **New registration**.
- 5. In the **Name** field enter the user-facing display name for the application. For example, Oracle-ERP-MicrosoftEmail.
- In the Supported account types section, select who can use this application or access this API.
- In the Redirect URI (optional) section, select Web from the drop-down field, and enter the redirect URL.

https://instance URL/icsapis/agent/oauth/callback

- 8. Click Register.
- 9. Get the Client ID.
 - a. In the left navigation pane, click **Overview**.
 - b. Copy the value in the **Application (client) ID** field.

Note down the value as it will be required while configuring the Microsoft Email connection from Oracle Integration.

- 10. Get the Client secret.
 - a. In the left navigation pane, click Certificates & secrets.
 - **b.** In the Client secrets section, click **New client secret**.
 - c. In the Add a client secret dialog:
 - i. Enter a description in the **Description** field.
 - ii. Select the expiration time of the app from the options available in the **Expires** field.
 - iii. Click Add.
 - **d.** In the resulting page, under the Client secrets section, copy the client secret value in the **Value** column.



Note down the value as it will be required while configuring the Microsoft Email connection from Oracle Integration.

- 11. Set the permissions and scopes for the application.
 - a. In the left navigation pane, click API permissions.
 - **b.** In the API permissions page, click **Add a permission**.
 - c. In the resulting Request API permissions dialog, click **Microsoft Graph**.
 - d. In the **Select permission** field, enter Mail.
 - **e.** Expand the Mail section, to see the available permissions. Select the following permissions:
 - Mail.ReadWrite
 - Mail.Send
 - f. Click Add permissions.

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the Install + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle ERP Cloud Connection
- Configure the Oracle Microsoft Email Connection
- Update Integration Properties

Configure the Oracle ERP Cloud Connection

- 1. On the Configuration Editor page, select **Oracle ERP Cloud Connection**.
- 2. Click Edit

The connection configuration page appears.

- 3. In the Connection Properties section, enter the Oracle ERP Cloud host name. For example: https://your domain name.fa.DC.oraclecloud.com.
- 4. In the Security section, enter the following details.

Field	Information to Enter
Security Policy	Select Username Password Token.



Field	Information to Enter
User Name	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.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle Microsoft Email Connection

- 1. On the Configuration Editor page, select **Oracle MicrosoftEmail Connection**.
- 2. Click Edit

The connection configuration page appears.

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

Field	Information to Enter
Client ID	Enter the Client ID obtained when you configured Microsoft Email. See Configure Microsoft Email.
Client Secret	Enter the Client Secret obtained when you configured Microsoft Email. See Configure Microsoft Email.
Scope	Enter the scopes URLs. See Configure Microsoft Email.
	You must include the offline_access scope along with the application-specific scopes. A blank space between each scope is also required. For example:
	https://outlook.office.com/mail.send https://outlook.office.com/ mail.readwrite offline_access

- 4. Click **Save**. If prompted, click **Save** for a second time.
- 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 Microsoft Email page.

7. Enter your Microsoft Email 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 Microsoft Email Connection window of Oracle Integration to test your connection.



In the Microsoft Email Connection window, click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

10. Click Back to return to the Configuration Editor page. Click Save if prompted.

Update Integration Properties

Update the integration properties with appropriate values.

The recipe's integration flow contains the following properties:

- **fromEmail**: This integration property holds the email address from which the order fulfillment notification emails are sent. Enter an email address of your choice.
- **fromName**: This integration property holds the name from which the order fulfillment notification emails are sent. Enter a name of your choice. For example, ERP Warehouse
- toEmail: This integration property holds the email address to which order fulfillment notifications and run time exception emails are sent. Enter an email address of your choice
- **toEmailName**: This integration property holds the name to which order fulfillment notifications and run time exception emails are sent. Enter a name of your choice.

For the procedure to update these integration properties, see Steps 7 to 9 in Override Design-Time Properties in an Integration.

Activate and Run the Recipe

After you've configured the connections and updated the integration property values, activate the recipe package and run it.

 On the Configuration Editor page, click Activate in the title bar. In the Activate Package dialog, click Activate again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration flow.

- 2. Run the recipe.
 - a. Log in to the Oracle ERP Cloud instance as a Sales user with the Order Management role to fulfill an order.



To fulfill an order, you must have a sales order created and released in Oracle ERP Cloud, and obtained its order number.

- **b.** Click the **Supply Chain Execution** tab from the navigator in the home page, and then select **Inventory Management**.
- c. Search for your order in the Inventory Management page.
 - Click the Quick Search icon on the right side of the page.
 - ii. In the resulting Search dialog:



- From the first field's drop-down list, select Shipments.
- From the second field's drop-down list, select **Order**.
- In the third field, enter your order number.
- Click Search.

Your order with details such as Shipment Status, Customer, and so on, is displayed on the Shipment page.

- **d.** Expand the **Lines** section, and in the Summary table:
 - i. Enter the quantity to be shipped in the **Shipped Quantity** column field.
 - Enter a number to track the shipment in the Tracking Number column field.
- e. Click Ship Confirm.

You get a confirmation that your shipment was confirmed. Click **OK**.

f. Click Save.

You've now successfully triggered the recipe.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. On the Configuration Editor page of the recipe, select the integration flow.
 - b. Click Run , then click Track Instances.
 - **c.** On the Track Instances page, you'll see the integration flow of the recipe being triggered and executing successfully.

The recipe now sends a notification email with the order details to the order contact's email address.

Check if you received an email with details of the order that was fulfilled in Oracle ERP Cloud.

Related Documentation

- Using the Oracle ERP Cloud Adapter with Oracle Integration
- Using the Microsoft Office 365 Outlook Adapter with Oracle Integration

Synchronize Inventory Items Between Oracle Product Hub Cloud and Oracle E-Business Suite

Use this recipe to create a corresponding inventory item in Oracle E-Business Suite when an item is created in Oracle Product Hub Cloud.



This recipe is available as **Oracle Product Hub Cloud** — **Oracle E-Business Suite | Synchronize Inventory Items** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

This recipe synchronizes inventory item records from Oracle Product Hub Cloud to Oracle E-Business Suite in real time. It uses the standard Oracle ERP Cloud Adapter and the Oracle E-Business Suite Adapter. To use the recipe, you must install the recipe package and configure the connections and other resources within the package.

When an inventory item is created in Oracle Product Hub Cloud (which is part of Oracle ERP Cloud), the integration flow of the recipe is triggered. The integration flow receives the inventory item details from Oracle Product Hub Cloud and creates a corresponding item in the PLM module of Oracle E-Business Suite, thereby synchronizing item records between the two systems.



- You can run this recipe for one inventory item at a time.
- Updating an item in Oracle Product Hub Cloud doesn't trigger the recipe.

System and Access Requirements

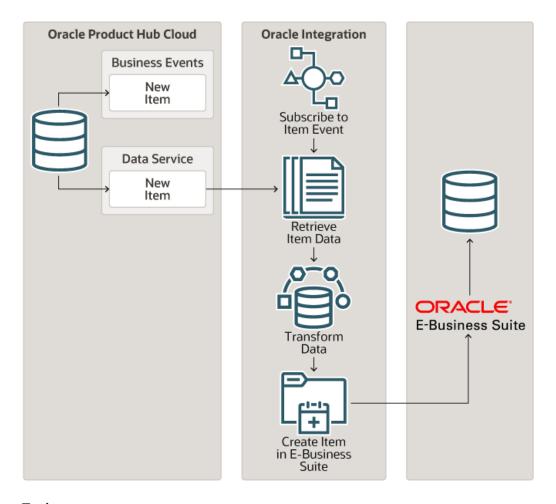
- Oracle Integration, Version 21.2.2.0.0 (210505.1400.40944) or higher
- Oracle E-Business Suite, Version 12.2.X
- Oracle ERP Cloud
- An account on Oracle E-Business Suite with the Administrator role
- Accounts on Oracle ERP Cloud with the Administrator role and the Product Data Steward role (in order to create new items)

Recipe Schema

This section provides an architectural overview of the recipe.

The integration flow of the recipe is triggered when an inventory item is created in Oracle Product Hub Cloud. It receives the inventory item data, transforms the data to the format supported by Oracle E-Business Suite, and finally creates a corresponding inventory item in Oracle E-Business Suite.





Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- · Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance and Oracle E-Business Suite instance in order to successfully connect to these external systems using Oracle Integration and achieve synchronization.

- 1. Configure Oracle ERP Cloud
- 2. Configure Oracle E-Business Suite

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud (or Oracle Product Hub 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.

- 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 account to connect to Oracle ERP Cloud from Oracle Integration.
- 2. Assign the following roles to the user account. For more information, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Oracle E-Business Suite

To access E-Business Suite from Oracle Integration, you must perform certain configurations on your E-Business Suite instance. In addition, you'll require a separate user account on E-Business Suite with necessary security grants.

Log in to your Oracle E-Business Suite instance as an **Administrator** and execute the following tasks.

- Create a user account for Oracle Integration. Make a note of the user name and
 password you set for the account. For the general procedure to create a new user
 account, see Create an Application User on Oracle E-Business Suite. (Skip the role
 assignment specified in the referenced topic as it's not applicable to this recipe.)
- 2. Perform the general configurations listed on the following page: Setup Tasks for Enabling the Oracle E-Business Suite Adapter. Here, make sure you provide the security grants to the user account created for Oracle Integration.
- 3. In addition to the REST services you deployed in the previous step, you must deploy the Catalog Item Maintenance (EGO_ITEM_PUB) REST service, and create a security grant on this service for the user account created for Oracle Integration.
 - a. Deploy the Catalog Item Maintenance service with the following details:
 - Enter plmsvc as the service alias.
 - Select the Process Item (convenience wrapper version) method.
 - **b.** After the service is deployed, create a grant on the **Process Item (convenience wrapper version)** method for the user account created for Oracle Integration.

For more information on deploying REST services and creating security grants, see Deploying REST Web Services and Managing Grants for Interfaces with Support for SOAP and REST Web Services in *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.



5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle ERP Cloud Connection
- Configure the Oracle E-Business Suite Connection
- Configure the Lookup Table

Configure the Oracle ERP Cloud Connection

- **1.** On the Configuration Editor page, select **Oracle ERP Cloud Connection**.
- 2. Click Edit

The connection configuration page appears.

- 3. In the Connection Properties section, enter the Oracle ERP Cloud host name. For example: https://your domain name.fa.DC.oraclecloud.com.
- 4. 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 on Oracle ERP Cloud. See Configure Oracle ERP Cloud.
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

 Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Oracle E-Business Suite Connection

- 1. On the Configuration Editor page, select **Oracle E-Business Suite Connection**.
- 2. Click Edit

The connection configuration page appears.

- 3. In the Connection Properties section, enter the URL of your Oracle E-Business Suite instance.
- 4. In the Security section, enter the following details.



Field	Information to Enter	
Username	Enter the user name of the account created for Oracle Integration on Oracle E-Business Suite. See Configure Oracle E-Business Suite.	
Password	Enter the password of the account created for Oracle Integration on Oracle E-Business Suite.	

- 5. Click Save. If prompted, click Save for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.
 - A confirmation message is displayed if your test is successful.
- 7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Lookup Table

Edit the lookup table to enter the necessary details.

- On the Configuration Editor page, select the lookup table and click Edit .
- 2. Against the **toEmail** field, enter the email address to which run-time exception emails are to be sent.
- 3. Against the **fromEmail** field, enter the email address from which run-time exception emails are to be sent.
- **4.** Against the **p_organization_code** field, enter the code of the organization in Oracle E-Business Suite with which to associate the new items created by this recipe.
- 5. Click Save. If prompted, click Save for a second time.
- 6. Click **Back** to return to the Configuration Editor page.

Activate and Run the Recipe

After you've configured the connections and the lookup table, activate the recipe package and run it.

- On the Configuration Editor page, click Activate in the title bar. In the Activate Package dialog, click Activate again.
 - A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration flow.
- 2. Run the recipe.
 - a. Log in to the Oracle ERP Cloud instance using a separate account, for example, as a user with the Product Data Steward role.
 - b. Click the **Product Management** tab on the home page, and then select **Product Information Management**.
 - On the resulting page, click Tasks from the right pane, and then click Create Item.
 - d. In the Create Item dialog:
 - i. Enter an organization.
 - ii. Leave the Create New radio button selected.
 - iii. Enter 1 in the **Number of Items** field.



Note:

For successful execution of the recipe, you must create a single item.

- iv. Select the required item class and template.
- v. Click **OK**.

 If you're shown a warning message, click **Yes**.
- e. On the Create Item page:
 - i. Enter a name and description for the item.
 - ii. From the Save drop-down list, click Save and Close.

You've now successfully created an item and triggered the recipe.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. On the Configuration Editor page of the recipe, select the integration flow.
 - b. Click Run , then click Track Instances.
 - c. On the Track Instances page:
 - You'll see the integration flow of the recipe being triggered and executing successfully.
 - The recipe now creates a new inventory item in Oracle E-Business Suite corresponding to the item created in Oracle ERP Cloud.
 - ii. Select the row of the integration instance that just executed, and click View Details.
 - The Activity Stream pane appears, in which you can view the ID of the inventory item created in Oracle E-Business Suite.
- 4. Log in to your Oracle E-Business Suite instance as an Administrator and check for the new item record.
 - a. On the home page, enter Master Items in the search field.
 - **b.** From the search results, select the Master Items entry corresponding to the role and organization you've configured in the recipe.
 - c. In the resulting dialog, on the Functions tab, select the Item Search entry and click Open.
 - d. In the Find Items dialog, against the Item Mask field, enter the name of the inventory item you created in Oracle ERP Cloud. Click Find.

Verify the item details displayed.

Related Documentation

- Using the Oracle ERP Cloud Adapter with Oracle Integration
- Using the Oracle E-Business Suite Adapter with Oracle Integration



Synchronize 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.



This recipe is available as **Oracle CX Sales and B2B Service** — **Oracle NetSuite | Synchronize Opportunities and Orders** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe synchronizes opportunities in Oracle CX Sales and B2B Service with 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 package and configure the connections and other resources within the package.

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 NetSuite. The subsidiary flows check if the particular customer and inventory item records exist in 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 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 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



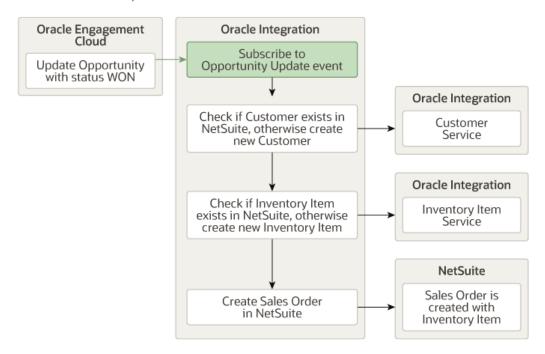
- An account on Oracle NetSuite with the Administrator role
- Accounts on Oracle CX Sales and B2B Service with the Administrator role and the Sales User role (in order to create and close opportunities)

Integration Schemas

This section describes the integration flows present in the recipe package.

Main Integration Flow - Create Sales Order in NetSuite

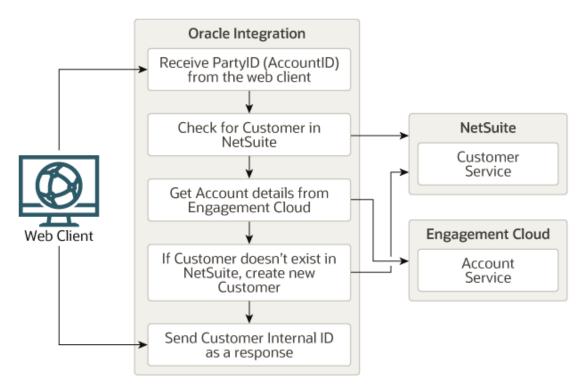
This integration flow is triggered when an opportunity is closed with the status *Won* in Oracle CX Sales and B2B Service. It receives the customer and inventory item details associated with the opportunity won, and it calls the subsidiary integration flows to fetch the corresponding customer and inventory item details from NetSuite. After it receives the details, the main flow creates a new sales order in NetSuite.



Subsidiary Integration Flow - Fetch Customer Record from NetSuite

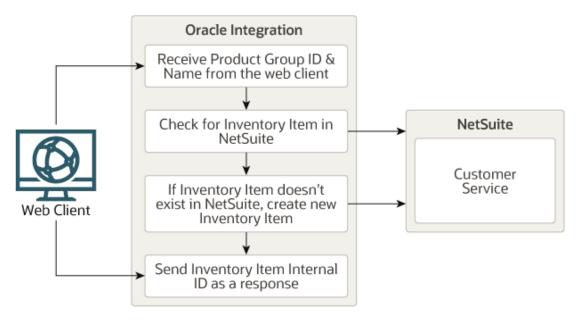
This integration flow is triggered by a REST service from the main integration flow. As a parameter, it receives the PartyId (AccountId) of the customer record associated with the opportunity won in Oracle CX Sales and B2B Service. Using this parameter, the flow checks if the corresponding customer record exists in 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 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 <code>prodgroupid</code> and <code>prodgroupname</code> 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 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 NetSuite and returns the internal ID of this record to the main flow.



Topics:

Before You Install the Recipe

- Install and Configure the Recipe
- Activate and Run the Recipe

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
- 2. Configure NetSuite

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 for Oracle 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.

- 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 Create
 an Integration User Account in Using the Oracle CX Sales and B2B Service
 Adapter with Oracle Integration.
 - 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*.
 - 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 Enable Event Subscriptions in the Oracle CX Sales and B2B Service in Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration.
 - 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 NetSuite

To access NetSuite from Oracle Integration, you must perform certain general configurations on your 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 NetSuite instance as an **Administrator** and execute the following tasks.

- 1. Perform the general configurations necessary to connect to 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
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 package to deploy and configure the integrations and associated resources.

- On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST Connection
- Configure the Oracle NetSuite Connection
- Configure the Oracle CX Sales and B2B Service Connection



Optionally, you can configure the following lookup table in the package.

• ORCL-BRT-EC_NS_SETTINGS: This lookup table contains a mapping of Oracle CX Sales and B2B Service country codes to NetSuite country codes. On your Oracle CX Sales and B2B Service instance, if you have an opportunity record associated with a country that isn't listed in the lookup table, add the mapping for that country code in the table. Additionally, this lookup table contains a default value for the internal ID of the NetSuite subsidiary to which the new sales orders will belong. To associate the new sales orders with the correct subsidiary on your NetSuite instance, enter the internal ID of the preferred subsidiary against the subsidiary field in the table. Also, against the ToEmailID field, you can enter an email ID on which you'd like to receive notifications related to the integration's execution.

Configure the Oracle REST Connection

Test and save the Oracle REST connection.

- 1. On the Configuration Editor page, select Oracle REST Connection.
- 2. Click Edit
- On the Oracle REST Connection page, click **Test** to ensure that your connection is correctly configured.

A confirmation message is displayed if your test is successful.



- 4. Click Save. If prompted, click Save for a second time.
- 5. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle NetSuite Connection

- 1. On the Configuration Editor page, select Oracle NetSuite Connection.
- 2. Click Edit .

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter
WSDL URL	Enter the NetSuite WSDL URL; for example, https://webservices.netsuite.com/wsdl/ <netsuite_application_version>/netsuite.wsdl.</netsuite_application_version>
	where <netsuite_application_version> is the version of the NetSuite application. For example, v2015_1_0. See Assemble the Oracle NetSuite WSDL URL.</netsuite_application_version>

4. 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.		
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.		
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.		



Note:

You must enter this information in capital letters

- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.



Configure the Oracle CX Sales and B2B Service Connection

- On the Configuration Editor page, select Oracle CX Sales and B2B Service Connection.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection Properties section, enter the Oracle CX Sales and B2B Service host name. For example: https:// customer chosen domain name.fa.DC.oraclecloud.com.

4. 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 Oracle Integration account created on the Oracle CX Sales and B2B Service instance. See Configure Oracle CX Sales and B2B Service.	
Password	Enter the password of the Oracle Integration account created on the Oracle CX Sales and B2B Service instance.	

- 5. Click Save. If prompted, click Save for a second time.
- 6. Click Test to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

 Click Back to return to the Configuration Editor page. Click Save again if prompted.

Activate and Run the Recipe

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

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.

A confirmation message is displayed informing that the integrations have been submitted for activation. Refresh the page to view the updated status of the integrations.

- 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:
 - 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.



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 execution of the integration flows in Oracle Integration.
 - a. On the Oracle Integration navigation pane, click **Home**, then **Monitoring**, then **Integrations**, and then **Tracking**.
 - **b.** On the Track Instances page, you'll see the subsidiary and main integration flows of the recipe package being triggered and executing successfully.

The recipe now creates a new sales order in Oracle NetSuite corresponding to the opportunity won in Oracle CX Sales and B2B Service.

- 4. Log in to your 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 NetSuite home page, select **Transactions**, then **Sales**, then **Enter Sales Orders**, and then **List**.
 - **b.** To view the customer record: On the NetSuite home page, select **Lists**, then **Relationships**, and then **Customers**.
 - c. To view the inventory item record: On the NetSuite home page, select **Lists**, then **Accounting**, and then **Items**.

Related Documentation

- Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration
- Using the Oracle NetSuite Adapter with Oracle Integration



Synchronize Supplier Updates Between Oracle Procurement Cloud and Oracle ATP

Use this recipe to synchronize supplier updates between Oracle Procurement Cloud and Oracle Autonomous Transaction Processing (ATP) database.



This recipe is available as **Oracle Procurement Cloud** — **Oracle ATP | Synchronize Supplier Updates** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe updates the Oracle ATP database in real time when supplier data (for example, products and services, transaction taxes, payments, addresses, sites, contacts) is updated in Oracle Procurement Cloud. The recipe contains two integration flows, one for receiving the supplier data updates from Oracle Procurement Cloud and then updating the same into an Apache Kafka topic, and another for writing the supplier data updates from the Apache Kafka topic into the Oracle ATP database table. The integrations use app-driven orchestration, and the standard Oracle ERP Cloud Adapter, Apache Kafka Adapter, and Oracle Autonomous Transaction Processing Adapter.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. Subsequently, you can activate the integrations. When supplier data is updated in Oracle Procurement Cloud, the first integration (*Oracle ERP Kafka Supplier Sync Producer*) is triggered. The integration flow receives the updates from Oracle Procurement Cloud and writes the updates into an Apache Kafka topic. This in turn triggers the second integration (*Oracle Kafka ATP Supplier Sync Consumer*) which consumes the updates from the Apache Kafka topic and writes the updates to the Oracle ATP database table, thereby synchronizing the supplier data updates between the systems.

System and Access Requirements

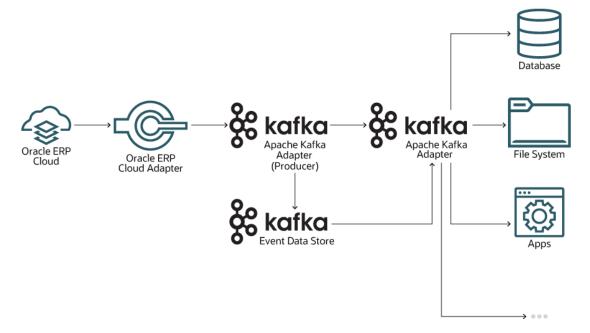
- Oracle Integration, Version 21.2.2.0.0 (210505.1400.40944)
- Oracle ERP Cloud
- Oracle ATP
- Apache Kafka
- An account in Oracle ERP Cloud with the Administrator and Procurement Manager roles
- An account in Apache Kafka with the Administrator role
- An account in Oracle ATP with the Administrator role
- An Oracle ATP database table with write access



Recipe Schema

This section provides an architectural overview of the recipe.

An event is raised in Oracle ERP Cloud (Oracle Procurement Cloud) when supplier data is updated. This triggers the first integration flow in Oracle Integration. The integration flow uses the Oracle ERP Cloud Adapter and the Apache Kafka Adapter (Producer) to get and write the updates to an Apache Kafka topic. This in turn triggers the second integration flow which uses the Apache Kafka Adapter (Consumer) and Oracle Autonomous Transaction Processing Adapter to write the supplier data updates from the Apache Kafka topic to an Oracle ATP database table.



Topics:

- · Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud, Apache Kafka, and Oracle Autonomous Transaction Processing (ATP) instances in order to successfully connect to these external systems using Oracle Integration and achieve synchronization.

- 1. Configure Oracle ERP Cloud
- 2. Configure Apache Kafka
- 3. Configure Oracle Autonomous Transaction Processing (ATP)



Configure Oracle ERP Cloud

To access the Oracle ERP Cloud (or Oracle Procurement 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.

- 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/privileges to the user account. For more information, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration*.
 - Integration Specialist
 - AttachmentsUser
 - FND MANAGE CATALOG SERVICE PRIV
 - ASM IMPLEMENTATION MANAGER DUTY

In addition to creating a separate user account and assigning roles/privileges to it, you have to enable supplier events from Oracle ERP Cloud. See Enable Supplier Events.

Enable Supplier Events

To get the required supplier data updates from Oracle Procurement Cloud, you need to subscribe to supplier Create/Update events in Oracle ERP Cloud.

The Supplier Create/Update events include the Supplier Number and SupplierID attributes in the output payload. Oracle Integration uses the event attributes to invoke the Supplier REST API and get the required supplier data updates from Oracle Procurement Cloud.

- 1. Log in to your Oracle ERP Cloud instance.
- Click My Enterprise from the navigator on the home page, and then click Feature Updates.
- 3. In the Features Overview page, select the **Available Features** tab.
- 4. Enter Suppliers in the functional area search field.
 - All available features that belong to the suppliers functional area get listed.
- 5. In the row for the feature Enabled Outbound Supplier Profile Integration Using Oracle Cloud, click **Enabled**.
- 6. In the resulting window, select the **Enable** check box for the feature Enable Outbound Supplier Profile Integration Using Oracle Cloud.
- 7. Click Done.



Configure Apache Kafka

Complete certain configuration tasks to successfully connect to Apache Kafka from Oracle Integration and achieve synchronization.

For more information, see Prerequisites for Creating a Connection in *Using the Apache Kafka Adapter with Oracle Integration*.

1. Get the Apache Kafka Cluster Bootstrap Server URL.

For more information, see Prerequisites for Creating a Connection in *Using the Apache Kafka Adapter with Oracle Integration*.

- 2. Install the Connectivity Agent
- 3. Create an Apache Kafka Topic

Install the Connectivity Agent

As Apache Kafka is hosted on-premise and is behind a firewall, you must install and configure the connectivity agent to allow Apache Kafka to interact with applications in the cloud.

Before you install the connectivity agent, review the following:

- System Requirements
- Connectivity Agent Restrictions

To install and configure the connectivity agent:

1. Create an agent group.

Note the agent group as you will need to select it while configuring the Apache Kafka connection from Oracle Integration.

Download and install the connectivity agent.

Create an Apache Kafka Topic

 From the command prompt execute the following command to create a topic named supplier_topic:

```
kafka-topics.bat --zookeeper 127.0.0.1:2181 --topic supplier_topic --
create -partitions 3 --replication-factor 1
```

You get a message that the topic is created.

2. To view the topic you just created, execute the following command:

```
kafka-topics.bat --zookeeper 127.0.0.1:2181 -list
```

A list of topics, including the supplier topic that you created, is displayed.



Configure Oracle Autonomous Transaction Processing (ATP)

Complete certain configuration tasks to successfully connect to Oracle ATP from Oracle Integration.

The following steps give an overview of the tasks you need to perform to obtain information and details for configuring the Oracle ATP connection property and security from Oracle Integration. For detail information, see Prerequisites for Creating a Connection in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration*.

1. Download the client credentials wallet.

This is a zip file containing the client security credentials. By default the file name is Wallet <databasename>.zip. You can save this file as any file name you want.

Get the database service name.

Note that the service name must be same as the one in the tnsnames.ora file in the client credential wallet.

3. Get the wallet password.

You'll be prompted to enter a wallet password while downloading the client credentials wallet. Note the password as you'll need it while configuring the Oracle ATP connection security from Oracle Integration.

In addition to the above configuration tasks, you need to set up a database table in your Oracle ATP instance to synchronize supplier information. See Create an Oracle ATP Database Table.

Create an Oracle ATP Database Table

- 1. Log in to your Oracle Autonomous Database (Transaction Processing) instance.
- 2. Navigate to the Service console, and in the left Autonomous Transaction Processing navigation menu, click **Development**.
- 3. Click Database Actions.
- 4. In the Database Actions page, under the Development section, click **SQL**.
- Using data definition language define and create your database object a
 database table named v_Supplier and specify the fields as shown in the table
 below.

Field Name	Data Type	Not Null	Primary Key
v_SupplierId	number	Yes	Yes
v_SupplierName	varchar2(20)	-	-
v_SupplierNumber	number	-	-
v_BusinessRelations hip	varchar2(20)	-	-
v_Status	varchar2(20)	-	-
v_DUNSNumber	int	-	-
v_TaxRegistrationNu mber	varchar2(20)	-	-



Field Name	Data Type	Not Null	Primary Key
v_TaxpayerId	varchar2(20)	-	-
v_TaxpayerCountryC ode	varchar2(20)	-	-
v_SupplierTypeCode	varchar2(20)	-	-
v_TaxOrganizationTy pe	varchar2(20)	-	-

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle ERP Cloud Connection
- Configure the Oracle Apache Kafka Connection
- Configure the Oracle ATP Connection
- Configure the Lookup Table

Configure the Oracle ERP Cloud Connection

- 1. On the Configuration Editor page, select **Oracle ERP Cloud Connection**.
- 2. Click Edit .

The connection configuration page appears.

- 3. In the Connection Properties section, enter the Oracle ERP Cloud host name. For example: https://your_domain_name.fa.DC.oraclecloud.com.
- 4. 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 on Oracle ERP Cloud. See Configure Oracle ERP Cloud.	
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.	

5. Click Save. If prompted, click Save for a second time.



6. Click **Test** to ensure that your connection is successfully configured.

A confirmation message is displayed if your test is successful.

 Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Oracle Apache Kafka Connection

- 1. On the Configuration Editor page, select Oracle Apache Kafka Connection.
- 2. Click Edit

The Oracle Apache Kafka Connection window appears.

3. In the **Bootstrap Servers** field, enter the Bootstrap Server URL you obtained earlier while configuring Apache Kafka.

See Configure Apache Kafka.

- 4. In the Security Policy field, select No Security Policy.
- 5. Select the agent group under the Agent Group section.
 - a. Click Configure Agents.
 - **b.** On the Select an Agent Group dialog, select the agent group that you had configured while installing the connectivity agent.

See Install the Connectivity Agent.

- c. Click Use.
- 6. On the Oracle Apache Kafka Connection window, click **Save**.
- Click Test to ensure that your connection is successfully configured. In the dialog that displays, click Test again. You get a confirmation message if your test is successful.
- 8. Click **Back** to return to the Configuration Editor page.

Configure the Oracle ATP Connection

- 1. On the Configuration Editor page, select Oracle ATP Connection.
- 2. Click Edit .

The Oracle ATP Connection window appears.

Under Connection Properties, in the Service Name field, enter the service name of your Oracle ATP instance.

See Configure Oracle Autonomous Transaction Processing (ATP).

- 4. Under the Security section, enter the following details:
 - In the Security Policy field, select JDBC Over SSL from the drop-down list.
 - In the Wallet field, click Upload to upload the credential wallet file that you obtained earlier.
 - Enter the password for your credential wallet file in the Wallet Password field.



- Enter your Oracle ATP account username in the Database Service Username field.
- Enter your Oracle ATP account password in the Database Service Password field.

See Configure Oracle Autonomous Transaction Processing (ATP).

- 5. Click **Save** on the connection window.
- 6. Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 7. Click **Back** to return to the Configuration Editor page.

Configure the Lookup Table

Edit the lookup table to enter the necessary details.

- 1. On the Configuration Editor page, select the lookup table and click **Edit** .
- 2. Edit a lookup. For example, against the **toEmail** field, enter the email address to which run time exception emails are to be sent.

You can update and modify the lookups as per your requirement.

- 3. Click Save. If prompted, click Save for a second time.
- 4. Click **Back** to return to the Configuration Editor page.

Activate and Run the Recipe

After you've configured the connections and the lookup table, activate the recipe package and run it.

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration flow.

- 2. Run the recipe.
 - **a.** Log in to the Oracle ERP Cloud instance as a user with the Procurement Manager role.
 - **b.** Click the **Procurement** tab from the navigator in the home page, and then select **Suppliers**.
 - c. In the resulting page, select **Manage Suppliers Task** and then search for a supplier you want to update.
 - d. In the supplier details, update one of the fields. For example, update the value in the **D-U-N-S Number** field.

You've now successfully triggered the recipe.

- 3. Monitor the execution of the integration flow in Oracle Integration.
 - a. Return to the Oracle Integration home page. In the left navigation pane, click **Monitoring**, then **Integrations**, and then **Tracking**.
 - **b.** On the Track Instances page, you'll see the integration flows of the recipe being triggered and executing successfully.



The recipe now updates the corresponding supplier's D-U-N-S Number in the Oracle ATP database table.

- 4. Log in to the Oracle ATP instance and check for the updated supplier data.
 - a. Open the V SUPPLIER SQL worksheet and execute the following query:

```
"SELECT * FROM V SUPPLIER"
```

b. In the query result, verify that the value in the supplier's v_dunsnumber column matches the value that was updated in Oracle Procurement Cloud.

If the value matches, then synchronization is achieved successfully.

Related Documentation

- Using the Oracle ERP Cloud Adapter with Oracle Integration
- Using the Apache Kafka Adapter with Oracle Integration
- Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration

Human Capital Management (HCM) Recipes

The Human Capital Management (HCM) recipes help you integrate applications related to recruiting, managing, and developing the employees in your company.

Topics:

- Export Employee Data from Oracle HCM Cloud to an Identity Management System
- Import Users from Oracle HCM Cloud to Okta
- Import Worker Records from an FTP Server to Oracle HCM Cloud
- Transform an Oracle HCM Cloud Extract File to an ANSI 834 Benefits File

Export Employee Data from Oracle HCM Cloud to an Identity Management System

Use this recipe to build an outbound integration to a customer identity management system when a new hire is created in Oracle HCM Cloud.



This recipe is available as **Oracle HCM Cloud** — **Identity Management** | **Sync Directories** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe demonstrates how to build an outbound integration to a customer identity management system when a new hire is created in Oracle HCM Cloud



Using the ATOM feed within the Oracle HCM Cloud Adapter, this recipe illustrates the ability to subscribe to New Hire notifications, build a full data set of profile data using additional REST calls, and then write the results in a different format to a third-party system.

System and Access Requirements

- Oracle Integration
- Oracle HCM Cloud, Release 13 or later
- FTP server
- DateTimeLib4 1.0.jar
- getEmployeeResponse.json
- newEmployeeFile.csv

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

PDF documentation and supporting files

Import Users from Oracle HCM Cloud to Okta

Use this recipe to create Okta users based on new hire information in Oracle HCM Cloud.



This recipe is available as **Oracle HCM Cloud** — **Okta | Generate Okta Users** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe demonstrates how to use Oracle Integration to create users in Okta based on the information for newly hired employees in Oracle HCM Cloud.

The integration performs the following actions:

- Retrieves the information for newly hired employees via the HCM Cloud ATOM feed.
- Enriches and transforms the data using additional REST calls.
- Generates the new users in the Okta access management system.

System and Access Requirements

- Oracle Integration
- Oracle HCM Cloud, Release 13 or later
- Okta instance



Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

· PDF documentation and supporting files

Import Worker Records from an FTP Server to Oracle HCM Cloud

Use this recipe to create workers in Oracle HCM Cloud based on hiring data from an external system.



This recipe is available as **Oracle HCM Cloud** — **FTP Server | Create Workers** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be errorfree. No support is provided for this recipe.

Overview

This recipe demonstrates how to use Oracle Integration to create workers in Oracle HCM Cloud. The integration handles different types of workers including permanent employees, contingent workers, and pending workers.

The integration performs the following actions:

- Reads a file from an external FTP server that contains data for new workers.
- Transforms and formats the file to generate the REST payload.
- Create the appropriate individuals with worker types within Oracle HCM Cloud.

System and Access Requirements

- Oracle Integration
- Oracle Fusion Cloud Human Capital Management, Release 13 or later
- FTP server
- Employee record payload files

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

PDF documentation and supporting files



Transform an Oracle HCM Cloud Extract File to an ANSI 834 Benefits File

Use this recipe to generate and format an ANSI 834 benefits provider input file, based on an Oracle HCM Cloud Extract file loaded from an FTP server.



This recipe is available as **Oracle HCM Cloud** — **FTP Server | Export Benefits ANSI 834** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

HCM users require the ability to synchronize HCM benefits data with third-party benefits providers. These providers often expect data to be published using the ANSI 834 EDI Enrollment Implementation Format, which is the standard file format in the United States for electronically exchanging health plan enrollment data between employers and health insurance carriers.

This recipe uses the data mapper capabilities of Oracle Integration to demonstrate how to transform a standard data extract from Oracle HCM Cloud into a Benefit Providers ANSI 834 data structure.

The integration uses Oracle Integration with an FTP server to complete the following actions:

- Reads the HCM exported data.
- Transforms and formats the file in Oracle Integration mapper.
- Produces a valid ANSI 834 file.
- Uploads the file to the FTP server.

For ease of setup, this recipe uses an FTP server to replicate the Oracle Fusion Cloud Human Capital Management instance. See the related documentation for details about how to use the Oracle HCM Cloud Adapter to retrieve files from an Oracle HCM Cloud instance via the HCM Extract tool.

Also, this recipe uses no encryption. In a production environment, it is critical that you encrypt data at every stage.

System and Access Requirements

- Oracle Integration
- FTP server

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

PDF documents and supporting files



Industries Recipes

The Industries recipes help you quickly integrate various industry-specific applications.

Topics:

Generate Oracle Hospitality Access Tokens

Generate Oracle Hospitality Access Tokens

Use this recipe to generate an access token from the Oracle Hospitality Integration Platform (OHIP).



This recipe is available as **Oracle Integration** — **Oracle Hospitality** | **Generate Access Tokens** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

You require an OHIP access token to call any Oracle Hospitality API. In addition, the tokens you obtain expire after sixty minutes. Using this recipe, you can automatically generate these tokens at a frequency of your choosing (for example, every thirty or sixty minutes), and use the tokens to seamlessly call the required Oracle Hospitality APIs.

To use the recipe, you must install the recipe package and configure the connection and other resources within the package. Subsequently, you can activate and run the integration flow of the package manually or specify an execution schedule for it. When triggered, the integration flow queries OHIP for an access token (using the credentials configured in the recipe's resources) and returns the token details in a log message. After an integration instance successfully executes, you can view the log message containing the token through the Activity Stream.

System Requirements

Oracle Integration, Version 21.2.1.0.0 or higher

Assumptions

- You have created an integration user account on the Oracle Hospitality Shared Security Domain (SSD) for the required hotel's tenant/chain, and possess a valid Integration ID and Integration Key for the account. See Partner Self-Registration Guide on the Oracle Hospitality Shared Security Domain (SSD) for OPERA documentation page.
- You have added a new environment in the Oracle Hospitality Developer Portal to access the hotel's environment, and obtained the Gateway URL, Client ID, and Client Secret of the environment added. See Adding an Environment and Viewing Environment Details in Oracle Hospitality Integration Platform User Guide.



 You have registered an application in the Oracle Hospitality Developer Portal (by selecting the right environment and subscribing to the required group of Oracle Hospitality APIs), and obtained the Application Key. See Registering an Application and Viewing the Application Key in Oracle Hospitality Integration Platform User Guide.

Topics:

- Install and Configure the Recipe
- Activate and Run the Recipe

Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the Install + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST Hospitality Connection
- Configure the Lookup Table

Configure the Oracle REST Hospitality Connection

- 1. On the Configuration Editor page, select **Oracle REST Hospitality Connection**.
- 2. Click Edit

The connection configuration page appears.

3. In the Connection Properties section, enter the following details.

Field	Information to Enter	
Connection Type	Leave REST API Base URL selected.	
Connection URL	Enter the Gateway URL obtained from the Oracle Hospitality Developer Portal. See Assumptions.	

4. In the Security section, enter the following details.

Field	Information to Enter	
Security Policy	Select Basic Authentication.	
Username	Enter the Client ID obtained from the Oracle Hospitality Developer Portal. See Assumptions.	
Password	Enter the Client Secret obtained from the Oracle Hospitality Developer Portal.	



- 5. Click **Save**. If prompted, click **Save** for a second time.
- Click Test to ensure that your connection is successfully configured.
 A confirmation message is displayed if your test is successful.
- Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Lookup Table

Edit the lookup table to enter the necessary details.

- 1. On the Configuration Editor page, select the lookup table and click **Edit** .
- 2. Against the **X-App-Key** field, enter the Application Key obtained from the Oracle Hospitality Developer Portal. See Assumptions.
- 3. Against the **ToEmailID** field, enter the email address to which run-time exception emails are to be sent.
- 4. Against the **FromEmailID** field, enter the email address from which run-time exception emails are to be sent.
- 5. Click **Save**. If prompted, click **Save** for a second time.
- 6. Click **Back ≤** to return to the Configuration Editor page.

Activate and Run the Recipe

After you've configured the connection and the lookup table, activate the recipe package and run it.

1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click **Activate** again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- 2. Update the integration property values.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Actions , then click Update Property Values.
 - **c.** In the Update Property Values dialog, enter the appropriate values for the following integration properties.

Click on a property row to expand it, and enter the appropriate value in the **New Value** field. Repeat for all property rows.

- GrantType: Enter "password".
- Username: Enter the Integration ID of the integration account created on Oracle Hospitality Shared Security Domain. See Assumptions.
- **Password:** Enter the Integration Key of the integration account created on Oracle Hospitality Shared Security Domain.
- d. Click Submit.



A confirmation message is displayed informing that the integration properties have been updated successfully.

- 3. Run the recipe.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Submit Now.
 - c. In the resulting dialog, click Submit Now again. Note that the Ad hoc request radio button is selected by default.

You've now successfully submitted the integration for execution. 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 execution of the integration flow and view the access token.
 - a. On the Configuration Editor page, select the integration flow.
 - b. Click Run , then click Track Instances.
 - c. On the Track Instances page:
 - You'll see the integration flow of the recipe being triggered and executing successfully.
 - The recipe now fetches an access token from OHIP.
 - Select the row of the integration instance that just executed, and click View Details.

The Activity Stream pane appears displaying the access token as a log message. You can use this access token to call the required Oracle Hospitality APIs. See Calling Oracle Hospitality APIs.

Related Documentation

Using the REST Adapter with Oracle Integration

Oracle Integration Starter Recipes

The starter recipes are basic integrations that required no additional setup. Use them to see how easy it is to activate, run, view results, and monitor an integration between endpoints.

Topics:

- Calculate Total Balance
- Concatenate Messages
- Echo Request
- · Get Incident Details
- Get Weather Data
- Increment the Sequence Variable within a Loop Integration
- Perform Mathematical Operations
- Get Stocks Data
- Say Hello World
- Say Hello World Invoke



- Say Hello World Map Data Invoke
- Send Files from Your Application by Email
- Transfer a File
- Use Multiple Verbs and Resources Invoke
- Work with Files on FTP Servers

Calculate Total Balance

Use this recipe to calculate the total balance for a group of credit cards using the for each action, and log the result to the activity stream.



This recipe is available as **Oracle Soap Library Calculator ForLoop** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

The integration in this recipe iterates through an array of credit cards using the for each action, and calculates the total balance.

The integration receives an array of credit cards in the SOAP request, and then calculates and logs the total balance for 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.

The integration that results from installing this recipe is named: Oracle Soap Library Calculator ForLoop.

To run this integration the input you provide must use WS-Security UsernameToken authentication and specify a user, a password, a token, and a time stamp. Use the user and password that you use to access your Oracle Integration instance.

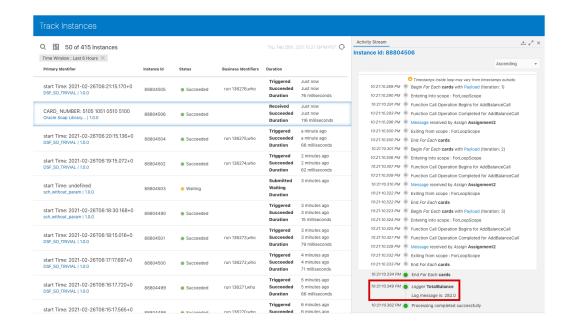
Input Example



```
<wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-</pre>
wss-soap-message-security-1.0#Base64Binary">1yzXqn9Ai17AyC/n+5pDpA==</wsse:Nonce>
            <wsu:Created>2021-02-26T18:19:00.372Z</wsu:Created>
         </wsse:UsernameToken>
      </wsse:Security>
   </soapenv:Header>
   <soapenv:Body>
      <aut:creditCards>
         <!--1 or more repetitions:-->
         <aut:CreditCard>
            <aut:CARD NUMBER>5105 1051 0510 5100</aut:CARD NUMBER>
            <aut:BALANCE>151</aut:BALANCE>
            <aut:LOAN AMOUNT>100</aut:LOAN AMOUNT>
            <aut:CREDIT RATE>4</aut:CREDIT RATE>
         </aut:CreditCard>
         <aut:CreditCard>
            <aut:CARD NUMBER>3782 8224 6310 005</aut:CARD NUMBER>
            <aut:BALANCE>94</aut:BALANCE>
            <aut:LOAN AMOUNT>80</aut:LOAN AMOUNT>
            <aut:CREDIT RATE>4</aut:CREDIT RATE>
         </aut:CreditCard>
         <aut:CreditCard>
            <aut:CARD NUMBER>3530 1113 3330 0000</aut:CARD NUMBER>
            <aut:BALANCE>37</aut:BALANCE>
            <aut:LOAN AMOUNT>20</aut:LOAN AMOUNT>
            <aut:CREDIT RATE>4</aut:CREDIT RATE>
         </aut:CreditCard>
      </aut:creditCards>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Example

This image shows the logged message in the activity stream. The total balance for this example is 282.



System and Access Requirements

Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Before you run the integration created with this recipe, you must configure the connection to the SOAP service:

To create a connection to the SOAP service, follow the steps described in Create a SOAP Adapter Connection in *Using the SOAP Adapter with Oracle Integration*.

Related Documentation

To learn more about the adapters used in this recipe, see *Using the SOAP Adapter with Oracle Integration*.

Concatenate Messages

Use this recipe to create and invoke a JavaScript callout activity to concatenate messages. This recipe uses the REST Adapter in an orchestrated integration.



This recipe is available as **Oracle Integration** — **REST | Concatenate Messages** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Description

This recipe demonstrates how to use a simple callout activity with a REST Adapter in an orchestrated integration. The REST Adapter is triggered when you specify a URL with two string query parameters. The JavaScript **Callout** function then concatenates these parameters and returns a string message as a response.

A logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor message status.

System Requirements

Oracle Integration

Related Documentation

Run the ConcatMessages Sample

Echo Request

Use this recipe to see a simple log and REST request-response trigger.



This recipe is available as **Oracle Integration** — **REST | Echo Request** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Description

This recipe demonstrates how you can use simple logging with a REST Adapter as a trigger in a request-response orchestrated integration.

The REST Adapter gets triggered when you specify a URL. A browser response is sent to you. A logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor message status.

System Requirements

Oracle Integration

Related Documentation

Run the Echo Sample



Get Incident Details

Use this recipe to get incident details from Oracle B2C Service for an incident ID and send the incident details to the caller as a response.



This recipe is available as **Oracle Service Cloud** — **REST | Get Incident Details** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Description

This recipe demonstrates how to get incident details from Oracle B2C Service for an incident ID and send the incident details to the caller as a response.

The REST Adapter is triggered when you specify a URL. The Oracle Service Cloud (RightNow) Adapter is invoked in the integration, and is configured to get incident details from Oracle B2C Service. The incident response is returned to you.

A logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor message status.

System Requirements

- Oracle B2C Service
- Oracle Integration

Related Documentation

Run the Incident Details from Service Cloud Sample

Get Weather Data

Use this recipe to get the weather data from an external source for a city and send the details to the caller as a response.



This recipe is available as **Oracle REST Get Weather DataService** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe demonstrates how to get the weather data from an external source for a particular zip code and send the weather details to the caller as a response.



To use the recipe, you must configure the Oracle REST Weather Connection by specifying the connection URL of a weather application (for example, https://

api.openweathermap.org/data/2.5/weather). Subsequently, you can trigger the first REST Adapter of the recipe by specifying a zip code and the API key for the connection URL. The second REST Adapter (which uses the REST connection you configured) is then invoked in the integration, and it fetches the weather details from the specified connection URL. The response containing the weather data is returned to you.

A logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor the message status.

You can invoke the REST API using the following command to get the weather data through the recipe:

```
GET: https://host:port/ic/api/integration/v1/flows/rest/ORCL-R-
REST GET WEATHER DATA/1.0/weather?zip={zip code}&appid={api key}
```

Example Response Payload:

```
"coord": {
    "lon": -96.9905,
    "lat": 32.8653
},
"weather": [
    {
        "id": 804,
        "main": "Clouds",
        "description": "overcast clouds",
        "icon": "04d"
    }
],
"base": "stations",
"main": {
    "temp": 269.41,
    "feels like": 262.04,
    "temp min": 269.15,
    "temp max": 269.82,
    "pressure": 1024,
    "humidity": 86
"visibility": 10000,
"wind": {
    "speed": 6.69,
    "deg": 350
},
"clouds": {
    "all": 90
},
"dt": 1613058731,
"sys": {
    "type": 1,
    "id": 3807,
    "country": "US",
```



System and Access Requirements

- Oracle Integration
- A weather application's connection URL and API key

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

To use this recipe, you must configure the Oracle REST Weather Connection. You must supply the following information when you configure this connection:

Connection Name	Parameters Connection Properties	
Oracle REST Weather Connection		
	 Connection Type: REST API Base URL. 	
	 Connection URL: The connection URL of a weather application (for example, https://api.openweathermap.org/data/2.5/ weather). 	
	Security	
	 Security Policy: API Key Based Authentication. 	
	 API Key: The API key for the connection URL. 	
	For more details, see Create a REST Adapter Connection.	

Related Documentation

Using the REST Adapter with Oracle Integration

Increment the Sequence Variable within a Loop Integration

Use this recipe to increment the sequence or counter variable within a data mapper while processing data in batches.



This recipe is available as **Oracle Increment Variable Inside Loop** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

The Increment Counter Variable recipe uses a REST Adapter to fetch or receive data in batches from an external application. This recipe enables you to increment the sequence or counter variable within a data mapper, using the <code>position()</code> function, for each batch of data that's processed.

You can invoke the REST API using the following command to post data to the recipe:

```
POST: https://host:port/ic/api/integration/v1/flows/rest/ORCL-R-
INCREM VARIAB INSIDE LOOP/1.0/ITER
```

Example Input Payload:

Example Output Payload:

```
{
    "Accounting": [
            "Seq": "1",
            "Employee": {
                "firstName": "John1",
                "lastName": "Doe1",
                "age": 23
        },
            "Seq": "2",
            "Employee": {
                "firstName": "John2",
                "lastName": "Doe2",
                "age": 33
            }
        },
            "Seq": "3",
            "Employee": {
                "firstName": "John3",
                "lastName": "Doe3",
                "age": 44
            }
        },
            "Seq": "4",
            "Employee": {
                "firstName": "John4",
                "lastName": "Doe4",
```



```
"age": 53
}
},
{
    "Seq": "5",
    "Employee": {
        "firstName": "John5",
        "lastName": "Doe5",
        "age": 63
}
}
],
"errorOccurred": "false"
}
```

System and Access Requirements

Oracle Integration

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Related Documentation

- Using the REST Adapter with Oracle Integration
- Using the Oracle Mapper with Oracle Integration.

Perform Mathematical Operations

Use these recipes to perform basic arithmetic operations on the input data and handle faults encountered during the division operation.



These recipes are available in the **Oracle Integration** — **SOAP | Calculator Operations** package in the Integration Store. Oracle provides these recipes as samples only. These recipes are meant only for guidance, and are not warranted to be error-free. No support is provided for these recipes.

Overview

Using these recipes, you can:

- Perform a specific arithmetic operation (such as, addition, subtraction, multiplication, and division) on the input data using a Switch action. To enable you to perform these operations, a JavaScript library (Oracle-Library-Calculator) is included in the package.
- Perform a division operation on the input data, and catch or handle faults encountered during the operation.



Description

SOAP Library Calculator Switch

Performs any arithmetic operation (addition, subtraction, multiplication, and division) on the input data.

As input to the integration, you must provide two input values and specify the necessary value for the Switch action, indicating which operation to perform. The integration returns the output of the specified operation as the response.

Example Input Payload:

Example Output Payload:

```
<env:Body>
```

```
<AnyWithOptionResponse xmlns:wsa="http://</pre>
schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/
encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/
wsdl/" xmlns:wsp="http://schemas.xmlsoap.org/ws/
2004/09/policy" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" xmlns:msc="http://
schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/
addressing/policy" xmlns:nstrgmpr="http://Example.org"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://Example.org">
         <nstrgmpr:result>48.0</nstrgmpr:result>
      </AnyWithOptionResponse>
   </env:Body>
```



Description

SOAP Library Division Calculator

Performs the division operation on the input data.

As input to the integration, you must specify the dividend as the first input and the divisor as the second input. The integration returns the output of the operation as the response.

Example Input Payload:

Example Output Payload:

```
<DivideResponse xmlns:wsa="http://</pre>
schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:ns0="http://www.oracle.com/XSL/Transform/java/
oracle.tip.dvm.LookupValue" xmlns:ns1="http://
www.oracle.com/XSL/Transform/java/
com.bea.wli.sb.resources.icsxpathfunctions.ICSInstanceT
rackingFunctions" xmlns:soapenc="http://
schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://
schemas.xmlsoap.org/wsdl/" xmlns:wsp="http://
schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:nsmpr0="http://www.oracle.com/2014/03/ic/
integration/metadata" xmlns:dvm="http://
www.oracle.com/XSL/Transform/java/
com.bea.wli.sb.functions.dvm.DVMFunctions"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:orajs10="http://www.oracle.com/XSL/Transform/
java/
oracle.tip.pc.services.functions.JsExecutor xpath428770
522" xmlns:orajs11="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath151171
120" xmlns:wsaw="http://www.w3.org/2006/05/addressing/
wsdl" xmlns:wsa10="http://www.w3.org/2005/08/
addressing" xmlns:orajs16="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath126454
0111" xmlns:orajs17="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath908143
547" xmlns:orajs18="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath889320
555" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/
soap12/" xmlns:orajs19="http://www.oracle.com/XSL/
```



Description

Transform/java/ oracle.tip.pc.services.functions.JsExecutor_xpath226366 833" xmlns:orajs12="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath754435 699" xmlns:orajs9="http://www.oracle.com/XSL/Transform/ oracle.tip.pc.services.functions.JsExecutor xpath137275 3807" xmlns:orajs13="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath248742 258" xmlns:orajs7="http://www.oracle.com/XSL/Transform/ java/ oracle.tip.pc.services.functions.JsExecutor xpath591907 397" xmlns:orajs14="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath213040 8426" xmlns:orajs15="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath178652 5844" xmlns:orajs8="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath263876 15" xmlns:orajs5="http://www.oracle.com/XSL/Transform/ iava/ oracle.tip.pc.services.functions.JsExecutor xpath132287 2458" xmlns:orajs6="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath917958 894" xmlns:orajs3="http://www.oracle.com/XSL/Transform/ java/ oracle.tip.pc.services.functions.JsExecutor xpath86288" xmlns:orajs4="http://www.oracle.com/XSL/Transform/ iava/ oracle.tip.pc.services.functions.JsExecutor xpath199629 1323" xmlns:orajs1="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath729152 639" xmlns:orajs2="http://www.oracle.com/XSL/Transform/ oracle.tip.pc.services.functions.JsExecutor xpath161156 5945" xmlns:orajs0="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath503065 307" xmlns:msc="http://schemas.microsoft.com/ws/ 2005/12/wsdl/contract" xmlns:wsap="http:// schemas.xmlsoap.org/ws/2004/08/addressing/policy" xmlns:nstrgmpr="http://Example.org" xmlns:soap="http:// schemas.xmlsoap.org/wsdl/soap/" xmlns="http:// Example.org">



Integration Name	Description	
	<pre><nstrgmpr:result>20.0</nstrgmpr:result> </pre>	
	\/ bividekesponse/	



Description

SOAP Library Global FaultHandler

Performs the division operation on the input data, and uses a predefined, global fault handler to catch errors during the operation.

For example, when the input for the divisor is *zero*, the operation results in a fault. Using a global fault handler, the integration flow returns the fault details as a response.

As input to the recipe, you must specify the dividend as the first input and the divisor as the second input. The integration returns the output of the operation or fault details as the response.

Example Input Payload:

Example Output Payload:

```
<DivideResponse xmlns:wsa="http://</pre>
schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:ns0="http://www.oracle.com/XSL/Transform/java/
oracle.tip.dvm.LookupValue" xmlns:plnk="http://
schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:ns1="http://www.oracle.com/XSL/Transform/java/
com.bea.wli.sb.resources.icsxpathfunctions.ICSInstanceT
rackingFunctions" xmlns:soapenc="http://
schemas.xmlsoap.org/soap/encoding/" xmlns:wsp="http://
schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:nsmpr0="http://www.oracle.com/2014/03/ics/fault"
xmlns:dvm="http://www.oracle.com/XSL/Transform/java/
com.bea.wli.sb.functions.dvm.DVMFunctions"
xmlns:nsmpr1="http://www.oracle.com/2014/03/ic/
integration/metadata" xmlns:wsx="http://
schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:orajs30="http://www.oracle.com/XSL/Transform/
java/
oracle.tip.pc.services.functions.JsExecutor xpath908143
547" xmlns:orajs31="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath889320
555" xmlns:wsaw="http://www.w3.org/2006/05/addressing/
wsdl" xmlns:wsa10="http://www.w3.org/2005/08/
addressing" xmlns:orajs9="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath119774
4749" xmlns:orajs7="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath605343
```



Description

633" xmlns:orajs8="http://www.oracle.com/XSL/Transform/ oracle.tip.pc.services.functions.JsExecutor xpath185128 3090" xmlns:orajs5="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath188653 5281" xmlns:orajs6="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath640595 824" xmlns:orajs3="http://www.oracle.com/XSL/Transform/ java/ oracle.tip.pc.services.functions.JsExecutor xpath107305 9868" xmlns:orajs4="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath102406 8758" xmlns:orajs1="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor_xpath729152 639" xmlns:orajs2="http://www.oracle.com/XSL/Transform/ java/ oracle.tip.pc.services.functions.JsExecutor xpath149642 6531" xmlns:orajs0="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor_xpath503065 307" xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/ addressing/policy" xmlns:soap="http:// schemas.xmlsoap.org/wsdl/soap/" xmlns:orajs20="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath591907 397" xmlns:orajs21="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath263876 15" xmlns:orajs22="http://www.oracle.com/XSL/Transform/ iava/ oracle.tip.pc.services.functions.JsExecutor xpath137275 3807" xmlns:orajs27="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath213040 8426" xmlns:orajs28="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath178652 5844" xmlns:orajs29="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath126454 0111" xmlns:orajs23="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath428770 522" xmlns:orajs24="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath151171



Description

```
120" xmlns:orajs25="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath754435
699" xmlns:orajs26="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath248742
258" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:orajs10="http://www.oracle.com/XSL/Transform/
oracle.tip.pc.services.functions.JsExecutor xpath481947
08" xmlns:orajs11="http://www.oracle.com/XSL/Transform/
oracle.tip.pc.services.functions.JsExecutor xpath206724
5588" xmlns:orajs16="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath114692
5263" xmlns:orajs17="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath199629
1323" xmlns:orajs18="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath132287
2458" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/
soap12/" xmlns:orajs19="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath917958
894" xmlns:orajs12="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath161156
5945" xmlns:orajs13="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath182264
4184" xmlns:orajs14="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath133935
528" xmlns:orajs15="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath86288"
 xmlns:nstrgdfl="http://xmlns.oracle.com/procmon"
xmlns:ns28="http://xmlns.oracle.com/ics/tracking/
ics tracking context.xsd" xmlns:msc="http://
schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:nstrgmpr="http://Example.org" xmlns="http://
Example.org">
         <nstrgmpr:result>Review JavaScript error(s)
and fix them:
error: null@18 -> DivideByZeroError
   at: function=divide, loc=js/TestAPILibrary 1.0/
calc.js, line=18</nstrgmpr:result>
      </DivideResponse>
```



Description

SOAP Library Div Inscope FaultHandler

Performs the division operation on the input data, and uses an in-scope fault handler to catch and handle errors during the operation.

For example, when the input for the divisor is *zero*, the operation results in a fault. The in-scope fault handler in the integration flow handles the fault by interchanging the values of the divisor and the dividend. The integration flow then completes the division operation successfully and returns the result as a response.

As input to the integration, you must specify the dividend as the first input and the divisor as the second input. The integration returns the output of the operation or the output of the modified operation (in case the operators are interchanged due to a fault) as the response.

Example Input Payload:

Example Output Payload:

```
<DivideResponse xmlns:wsa="http://</pre>
schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:ns0="http://www.oracle.com/XSL/Transform/java/
oracle.tip.dvm.LookupValue" xmlns:ns1="http://
www.oracle.com/XSL/Transform/java/
com.bea.wli.sb.resources.icsxpathfunctions.ICSInstanceT
rackingFunctions" xmlns:soapenc="http://
schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://
schemas.xmlsoap.org/wsdl/" xmlns:wsp="http://
schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:nsmpr0="http://www.oracle.com/2014/03/ic/
integration/metadata" xmlns:dvm="http://
www.oracle.com/XSL/Transform/java/
com.bea.wli.sb.functions.dvm.DVMFunctions"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:orajs10="http://www.oracle.com/XSL/Transform/
java/
oracle.tip.pc.services.functions.JsExecutor xpath428770
522" xmlns:orajs11="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath151171
120" xmlns:wsaw="http://www.w3.org/2006/05/addressing/
wsdl" xmlns:wsa10="http://www.w3.org/2005/08/
addressing" xmlns:orajs16="http://www.oracle.com/XSL/
Transform/java/
oracle.tip.pc.services.functions.JsExecutor xpath126454
0111" xmlns:orajs17="http://www.oracle.com/XSL/
```



Integration Name

Description

Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath908143 547" xmlns:orajs18="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath889320 555" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/ soap12/" xmlns:orajs19="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath226366 833" xmlns:orajs12="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath754435 699" xmlns:orajs9="http://www.oracle.com/XSL/Transform/ oracle.tip.pc.services.functions.JsExecutor xpath137275 3807" xmlns:orajs13="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath248742 258" xmlns:orajs7="http://www.oracle.com/XSL/Transform/ java/ oracle.tip.pc.services.functions.JsExecutor xpath591907 397" xmlns:orajs14="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath213040 8426" xmlns:orajs15="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath178652 5844" xmlns:orajs8="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath263876 15" xmlns:orajs5="http://www.oracle.com/XSL/Transform/ iava/ oracle.tip.pc.services.functions.JsExecutor xpath132287 2458" xmlns:orajs6="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath917958 894" xmlns:orajs3="http://www.oracle.com/XSL/Transform/ iava/ oracle.tip.pc.services.functions.JsExecutor xpath86288" xmlns:orajs4="http://www.oracle.com/XSL/Transform/ iava/ oracle.tip.pc.services.functions.JsExecutor xpath199629 1323" xmlns:orajs1="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor xpath729152 639" xmlns:orajs2="http://www.oracle.com/XSL/Transform/ oracle.tip.pc.services.functions.JsExecutor xpath161156 5945" xmlns:orajs0="http://www.oracle.com/XSL/ Transform/java/



oracle.tip.pc.services.functions.JsExecutor xpath503065
307" xmlns:msc="http://schemas.microsoft.com/ws/ 2005/12/wsdl/contract" xmlns:wsap="http:// schemas.xmlsoap.org/ws/2004/08/addressing/policy" xmlns:nstrgmpr="http://Example.org" xmlns:soap="http:// schemas.xmlsoap.org/wsdl/soap/" xmlns="http:// Example.org">

System and Access Requirements

Oracle Integration

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

To use the recipes in the Oracle Integration — SOAP | Calculator Operations package, you must configure the Oracle SOAP Connection. You only need to configure this connection once. All the recipes in the package use the same connection.

You must supply the following information when you configure this connection.

Parameters
Connection Properties
 WSDL URL: Specify the WSDL URL to receive the input data. For more details, see Create a SOAP Adapter Connection in Using the SOAP Adapter with Oracle Integration.

Related Documentation

Using the SOAP Adapter with Oracle Integration

Get Stocks Data

Use this recipe to get the stock-related data of a company from an external source and send the details to the caller as a response.



This recipe is available as **Oracle REST Stock Service** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

This recipe demonstrates how to get the stock-related data of a particular company on a certain date from an external source and send the stock details to the caller as a response.

To use the recipe, you must configure the Oracle REST Get Stock Quote Connection by specifying the connection URL of a stocks application (for example, https://api.polygon.io/vl/open-close). Subsequently, you can trigger the first REST Adapter of the recipe by specifying a company symbol, date, and the API key for the connection URL. The second REST Adapter (which uses the REST connection you configured) is then invoked in the integration, and it fetches the stock details from the specified connection URL. The response containing the stock-related data is returned to you.

A logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor the message status.

You can invoke the REST API using the following command to get the stock-related data through the recipe:

```
GET: https://host:port/ic/api/integration/v1/flows/rest/ORCL-R-
REST_STOCK_SERVICE/1.0/v1/open-close?
symbol={company symbol}&date={date}&apiKey={api key}
```

Example Response Payload:

```
{
    "status": "OK",
    "from": "2020-09-09",
    "symbol": "IBM",
    "open": 122.13,
    "high": 123.7,
    "low": 121.29,
    "close": 122.26,
    "volume": 3875357,
    "afterHours": 122.7,
    "preMarket": 121.95
}
```

System and Access Requirements

- Oracle Integration
- A stocks application's connection URL and API key

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

To use this recipe, you must configure the Oracle REST Get Stock Quote Connection. You must supply the following information when you configure this connection.



Connection Name	Parameters	
Oracle REST Get Stock Quote Connection	Connection Properties	
Connection	 Connection Type: REST API Base URL. 	
	 Connection URL: The connection URL of a stocks application (for example, https://api.polygon.io/v1/open-close). 	
	Security	
	 Security Policy: API Key Based Authentication. 	
	 API Key: The API key for the connection URL. 	
	For more details, see Create a REST Adapter Connection.	

Related Documentation

Using the REST Adapter with Oracle Integration

Say Hello World

Use this recipe to see how you can use simple log and email notification actions with a REST Adapter in an orchestrated integration.



This recipe is available as **Oracle Integration** — **REST | Say Hello World** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Description

This recipe demonstrates how to use simple log and email notification actions with a REST Adapter in an orchestrated integration. The REST Adapter gets triggered when you specify a URL.

The integration has a switch activity with two rules is defined.

- If you provide both an email address and a name when triggering the REST Adapter, then a browser response and email response are sent to you.
- If you provide only a name when triggering the REST Adapter, then only a browser response is sent to you.

With either switch rule, a logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor message status.

System Requirements

Oracle Integration

Related Documentation

Run the Hello World Sample



Say Hello World Invoke

Use this recipe to call and invoke a simple REST web service in an orchestrated integration. Simple logging is also used.



This recipe is available as **Oracle Integration** — **REST | Say Hello World Invoke** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Description

This recipe demonstrates how to call and invoke a simple REST web service in an orchestrated integration. The REST Adapter is triggered when you specify a URL. The Hello World REST Adapter that you previously used in the Say Hello World recipe is invoked in the integration. The name and email address response are sent to you in JSON format.

A logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor message status.

System Requirements

Oracle Integration

Related Documentation

Run the Hello World Invoke Sample

Say Hello World Map Data Invoke

Use this recipe to call and invoke a simple REST web service using the Map Data integration pattern.



This recipe is available as **Oracle Integration** — **REST | Say Hello World Map Data Invoke** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Description

This recipe demonstrates how to call and invoke a simple REST web service using the Map Data integration pattern. The REST Adapter is triggered when you specify a URL. The Hello World Invoke REST Adapter is invoked in the integration. The name and email address response are sent to you in JSON format.



A logging message is created and logged to the activity stream for viewing.

You can also track the integration and monitor message status.

System Requirements

Oracle Integration

Related Documentation

Run the Hello World Map Data Invoke Sample

Send Files from Your Application by Email

Use this recipe to send files from an external application to a specified email address.



This recipe is available as **REST** — **Oracle Integration | Send Files by Email** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

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

To use the recipe, you must install the recipe package and configure the connection and other resources within the package. Subsequently, you can activate the integration flow of the package and send a REST request containing a file to the integration's endpoint URL from an external application. When triggered, the integration flow sends the file received to the specified email address as an attachment. If your REST request doesn't contain a file, the integration flow isn't triggered and the email message isn't sent.



You can send only one file at a time with this recipe.

System and Access Requirements

Oracle Integration, Version 21.2.1.0.0 or higher

Topics:

- Install and Configure the Recipe
- · Activate and Run the Recipe



Install and Configure the Recipe

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

- 1. On the Oracle Integration home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.
- 3. Find the recipe package that you want to install.
- 4. Select the package, and then click the **Install** + icon.
- 5. After the package is installed, click the **Configure** icon on the recipe card to configure the resources deployed by the package.

The Configuration Editor page opens, displaying all the resources of the recipe package. Configure the following resources before you activate and run the recipe.

- Configure the Oracle REST Trigger Connection
- · Configure the Lookup Table

Configure the Oracle REST Trigger Connection

Test and save the Oracle REST Trigger connection.

- 1. On the Configuration Editor page, select Oracle REST Trigger Connection.
- 2. Click Edit
- On the Oracle REST Trigger Connection page, click Test to ensure that your connection is correctly configured.

A confirmation message is displayed if your test is successful.

- 4. Click Save. If prompted, click Save for a second time.
- 5. Click Back to return to the Configuration Editor page. Click Save again if prompted.

Configure the Lookup Table

Edit the lookup table and enter the recipient email address to which you want the files sent.

- 1. On the Configuration Editor page, select the lookup table and click **Edit**
- Against the SenderEmail and RecipientEmail fields, enter the corresponding email addresses.
- 3. Click Save. If prompted, click Save for a second time.
- 4. Click **Back** to return to the Configuration Editor page.

Activate and Run the Recipe

After you've configured the connection and the lookup table, activate the recipe package and run it.



1. On the Configuration Editor page, click **Activate** in the title bar. In the Activate Package dialog, click Activate again.

A confirmation message is displayed informing that the integration has been submitted for activation. Refresh the page to view the updated status of the integration.

- Run the recipe from an external application.
 - a. On the Configuration Editor page, select the integration and click **Run** .



- b. On the resulting pop-up dialog, click the **Metadata URL** link.
- From the new tab 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 required file. Upload the file in the POST request's **Body** field.

You'll receive the status code 202 as a response in your application, which indicates that Oracle Integration has accepted your request. When the integration flow receives the request, it sends the file to the recipient email address specified in the lookup table.

- 3. Test the recipe in Oracle Integration.
 - a. On the Configuration Editor page, select the integration.
 - b. Click Run , then click Test.

The page to test the integration with a sample file is displayed.

- c. In the Request section, click **Body** and upload a file from your computer.
- d. Click Test.

The integration now sends the file you uploaded to the recipient email address specified in the lookup table. The Activity Stream pane appears displaying the status of the integration instance's execution.

- e. In the Response section of the test page, you should see the status as 202 Accepted.
- Log in to your email account and check the inbox for the email message from Oracle Integration.

The email contains the required file as an attachment.

Related Documentation

Using the REST Adapter with Oracle Integration

Transfer a File

Use this recipe to read an opaque file from a "/" directory and write the file to an "/ upload" directory in a simple scheduled orchestration. After activating the integration,



you go to the Actions menu and select Submit now or Add Schedule to run the scheduled integration.



This recipe is available as **Oracle Integration** — **FTP | Transfer File** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Description

This recipe demonstrates how to read an opaque file from a

/

directory and write the file to an

/upload

directory in a scheduled orchestrated integration. An FTP Adapter reads the file from the

/

directory and another FTP Adapter writes the file to the

/upload

directory. An assign action is configured to assign variables for the file name and file size.

A logging message is created to indicate that the file name has been read. The message is logged to the activity stream for viewing.

You can also track the integration and monitor message status.

System Requirements

Oracle Integration

Related Documentation

Run the File Transfer Sample



Use Multiple Verbs and Resources Invoke

Use this recipe to create and invoke a REST web service with multiple HTTP verbs and resources.



This recipe is available as **Oracle Integration** — **REST | Use Multiple Verbs and Resources Invoke** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Description

This recipe demonstrates how to use multiple HTTP verbs and resources configured for the REST trigger (also called an invoke) of an integration. A single REST trigger can be configured to have multiple HTTP verbs such as GET, PUT, POST, PATCH, and DELETE.

A single REST trigger can also be configured to have multiple resources. However, this recipe configures the REST Adapter with a single resource endpoint of /emp and a sub-resource /emp/loc.

The configured operations are:

- modifyEmployee (PATCH)
- updateEmployee (PUT)
- getEmployee (GET)
- getEmployeeLocation (GET)
- deleteEmployee (DELETE)
- addEmployee (POST)

System Requirements

Oracle Integration

Related Documentation

Run the Multiple Verbs and Resources Invoke sample



Work with Files on FTP Servers

Use these recipes to work with files on FTP servers.



These recipes are available as **FTP Bulk Import** and **FTP Read-Write Files List** in the Integration Store. Oracle provides these recipes as samples only. These recipes are meant only for guidance, and are not warranted to be error-free. No support is provided for these recipes.

Overview

The FTP recipes use the FTP Adapter to read and write files to an FTP server. These recipes enable you to import bulk files and modify their content in Oracle Integration, or copy a list of files from one directory to another on an FTP server.

You can store the set of files to import or 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 Generation 2*.

Recipe Name	Description
FTP Bulk Import	Retrieves a zip file (a set of files) from an FTP server, modifies the content
	of each file, and writes the transformed zip file back to the server.



For the files to be successfully read and transformed, you must provide the schema files (comma-separated value (CSV) or .txt files) that contain the required data structures in the Stage File actions for reading and writing files.

FTP Read-Write Files List

Copies a list of files from one directory to another on an FTP server.

System and Access Requirements

- Oracle Integration
- A secure FTP (sFTP) server or File Server
- An FTP client to access the external sFTP server

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.



To use the collection of FTP recipes, you must configure the Oracle FTP Connection. You only need to configure this connection once. All the FTP recipes listed above use the same connection.

For prerequisites to configure the connection, see Prerequisites for Creating a Connection.

For details on configuring the FTP connection, see Create an FTP Connection.

Related Documentation

- Create an Integration to Import and Process Bulk Files
- Using File Server in Oracle Integration Generation 2

Social and Productivity Recipes

Explore the Social and Productivity Recipes to create integrations with applications that let you grow your presence on social media or make your routine tasks easier and simpler.

Topics:

- Export Google Gmail Attachments to an FTP Server
- Integrate with DocuSign
- Interact with Facebook
- Interact with LinkedIn
- Interact with Slack
- Interact with Twitter
- Post Slack Notifications for Completed SurveyMonkey Surveys
- Synchronize SurveyMonkey with Oracle NetSuite

Export Google Gmail Attachments to an FTP Server

Use this recipe to export email attachments from Google Gmail to an FTP server.



This recipe is available as **Google Gmail** — **FTP | Export Attachments** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe exports attachments that you receive in Google Gmail to an FTP server, according to a schedule specified in Oracle Integration. The recipe uses the standard Google Gmail Adapter and the standard FTP Adapter.



To use the recipe you must install the recipe package and configure the connections and other resources within the package. After you activate and run the recipe, the integration uses a schedule parameter to filter out already exported attachments, fetches the latest attachments from Google Gmail, and writes them to the specified location on the FTP server.

System and Access Requirements

- Oracle Integration Version 21.2.1.0.0 or higher
- A valid account on Google Gmail
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Before You Install the Recipe

Follow these steps to enable and configure other systems that this recipe needs to run:

- 1. Configure Gmail
- 2. Access Your FTP Server and Create a Directory

Configure Gmail

You must obtain the Gmail API credentials, that is the Client ID and Secret, for connecting to Gmail from Oracle Integration.

- Log in to your Google account and access the Google Developer console at https:// console.developers.google.com.
- On the Title bar of the Google Cloud Platform page, click the Select a project drop-down list.
- 3. Select an existing project to use or create a new project.
 - Note that if you create a new project, ensure that the project is selected from the dropdown list after saving the new project. You'll be able to view the project's dashboard only if you select the project.
- Click ENABLE APIS AND SERVICES. Alternatively, click Library in the left navigation pane.

The API Library page opens.

- 5. In the Search field enter Gmail API, and then select Gmail API from the search results.
- 6. On the Gmail API window, click **MANAGE**.
- 7. In the left navigation pane, click Credentials.
- 8. At the top of the Credentials window, click **CREATE CREDENTIALS** and then choose **OAuth client ID**.
- 9. In the Application type field, select Web Application from the drop-down list.
- 10. Enter a name for the OAuth Client ID in the Name field.



- 11. Scroll to the Authorized redirect URIs section and click ADD URI.
- 12. Enter the following URL in the URIs field.

https://OIC HOST:OIC SSL PORT/icsapis/agent/oauth/callback

Where

- OIC HOST is the Oracle Integration instance host name.
- OIC_SSL_PORT is the Oracle Integration SSL port.

13. Click CREATE.

A message confirms that the OAuth Client has been created.

14. In the resulting dialog, copy the Client ID and Client Secret.

Note the Client ID and Secret. You'll need these later while configuring the Google mail connection from Oracle Integration.

15. Click OK.

You return to the Credentials window, and the OAuth Client ID that you created is listed under the OAuth 2.0 Client ID section.



Before creating a Gmail connection, you must upload the trusted Gmail public certificate to Oracle Integration. The trusted Gmail public certificate can be downloaded from https://gmail.com. Rename the GoogleCertificate.txt file extension to .cer.

See Upload an SSL Certificate and Certificate Errors in *Using the Google Gmail Adapter with Oracle Integration*.

Access Your FTP Server and Create a Directory

Obtain a 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 (gmail-attachments) on the server into which the attachments from Gmail will be written. Note down the path of the directory.

Install and Configure the Recipe

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

- On the Oracle Integration Home page, scroll to the Accelerators & Recipes section.
- 2. Click Search All.

The list of available recipes appears.



3. Select the recipe you want to install, and click **Install** +.

A confirmation message informing that the recipe was successfully installed appears, and the recipe now displays the **Installed** tag.

- 4. Click **Configure** to configure the following resources using the **Configuration Editor**.
 - · Configure the Oracle Google Mail Connection
 - Configure the Oracle FTP Read Write Connection
 - Update Integration Property

Configure the Oracle Google Mail Connection

You must provide connection details such as the client ID and client secret to access Google APIs, then test your connection to ensure it works.

- 1. On the Configuration Editor page, select Oracle Google Mail Connection.
- 2. Click Edit

The Google Mail Connection window opens.

- 3. Under the Security section, enter the following details:
 - In the Client Id and Client Secret fields, enter the client ID and secret that you
 obtained while configuring Gmail.

See Configure Gmail.

- b. In the **Scope** field, enter https://www.googleapis.com/auth/gmail.modify.
- 4. Click Save.
- 5. Click Provide Consent.
 - In the resulting dialog, enter your Oracle Integration username and password, and click Sign in.
 - b. Log in to your Gmail account from the browser window that opens.
 - Follow the prompts till you see the access allowed page.
 Switch back to your Google Mail Connection window in Oracle Integration.
- 6. Click **Test** to ensure that your connection is successfully configured.

You get a confirmation message if your test is successful.

7. Click **Back** to return to the Configuration Editor page. Click **Save** again if prompted.

Configure the Oracle FTP Read Write Connection

You must configure connection details such as connection properties and log in credentials for your FTP server.

- 1. On the Configuration Editor page, select Oracle FTP Read Write Connection.
- 2. Click Edit

The FTP Connection window appears.



3. In the Connection 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	22
SFTP Connection	Select Yes from the list.

4. 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.

- 5. Click **Save**. If prompted, click **Save** for a second time to confirm.
- Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again.

You get a confirmation message if your test is successful.

Click Back to return to the Configuration Editor page. Click Save again if prompted.

Update Integration Property

You must update the integration property value so that the attachments from Gmail are exported to the right folder in the FTP server.

- On the Configuration Editor page, select the Oracle Gmail FTP Attachment Sync integration.
- 2. Click Actions , and then click Update Property Values.
- In the Update Property Values dialog, note the property that has been configured -Ftpfolderdestination.
- 4. Click Ftpfolderdestination, and in the New Value field, enter the path of the directory where the Gmail attachments will be written in the FTP server. For example, oracle-int/gmail-attachments.

See Access Your FTP Server and Create a Directory.

5. 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 updated the integration property, you can activate and run the recipe.

1. On the Configuration Editor page, click Activate.



2. In the Activate Package dialog, click Activate.

You get a confirmation message that the integration has been activated.

3. On the Configuration Editor page, click **Run** on the integration's row, and then select **Submit Now**.

The Schedule Parameters window appears.

 In the Schedule Parameters window, enter the current date (format: YYYY-MM-DD) in the New Value field. For example, 2021-07-20.

This filters out already exported attachments, and ensures that only the latest attachments in the Gmail account after the last export gets fetched.

5. Click Submit.

You've successfully submitted the integration for a test run.

Log in to the FTP server and check if the attachments from Gmail have been exported to the specified directory.

Related Documentation

- Using the Google Gmail Adapter with Oracle Integration
- Using the FTP Adapter with Oracle Integration

Integrate with DocuSign

Use these recipes to create a draft envelope in your DocuSign account and retrieve the status of an existing DocuSign envelope, respectively.



These recipes are available in the Integration Store as:

- REST DocuSign | Create Envelope
- REST DocuSign | Get Envelope Status

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

Overview

With these recipes, you can:

- Create a draft envelope containing a document (for example, a PDF document) in your DocuSign account. Subsequently, from your DocuSign account, you can edit this envelope and send it to the required recipients.
- Retrieve the status of an existing envelope in your DocuSign account.

To run the DocuSign recipes, you must configure the Oracle DocuSign Connection with the details necessary to access the DocuSign application. You can then trigger the recipes through a REST request and supply the necessary request payload.

• For the REST — DocuSign | Create Envelope recipe, you must supply the ID of the document you want to insert within a DocuSign envelope, the name of the document, and

the Base64 value for the document as the request payload. The DocuSign Adapter in the recipe is subsequently invoked, and it creates a draft envelope with your document in your DocuSign account. The response containing the new envelope ID is returned to you.

• For the REST — DocuSign | Get Envelope Status recipe, you must supply the ID of the DocuSign envelope of which you want the status as the request payload. The DocuSign Adapter in the recipe is subsequently invoked, and it fetches the status of the specified envelope. The response containing the status and the last status-change date and time is returned to you.

Integration Name	Description	REST Commands and Example Payloads
Create Envelope	Creates a draft envelope containing a document in your	REST API Command:
	DocuSign account.	<pre>POST: https://<host:port>/ic/api/ integration/v1/flows/rest/ORCL-R- REST_DOCUSI_SEND_ENVELO/1.0/ createdocusign</host:port></pre>
		Example Request Payload:
		<pre>{ "documentId" : "12345", "name" : "new", "documentBase64" : "<base64_equivalent_of_a_pdf_file>" }</base64_equivalent_of_a_pdf_file></pre>
		Example Response Payload:
		<pre>{ "envelopeId": "cdf684b3-blaf-4d2a- a56e-4d3a146c5c08" }</pre>



Integration Name	Description	REST Commands and Example Payloads
Get Envelope Status	Retrieves the status of an existing envelope in your	REST API Command:
DocuSign account.		<pre>POST: https://<host:port>/ic/api/ integration/v1/flows/rest/ORCL-R- REST_DOCUSI_GET_ENVE_STAT/1.0/ envelopstatuschanges</host:port></pre>
		Example Request Payload:
		{ "envelopidId":"13e47618-b03f-4fac-ac4d-3fc0c9477311" }
		Example Response Payload:
		<pre>{ "status": "created", "statuschangedatetime": "2021-04-13T13:22:20.1730000Z" }</pre>

System and Access Requirements

- Oracle Integration
- DocuSign

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Configure the Oracle REST Connection and the Oracle DocuSign Connection present in the recipe package. You only need to configure these connections once. Both the DocuSign recipes use the same connections.

You must supply the following information when you configure these connections.

Connection Name	Parameters
Oracle REST Connection	The connection parameters are already configured for you. Test and save the connection.



Connection Name	Parameters	
Oracle DocuSign Connection	Security	
	 Client ID (Integrator Key): The App/Integration Key of the integration created on DocuSign. 	
	 Client Secret: The Secret Key of the integration created on DocuSign. 	
	• Scope: The list of scopes for which you want to grant permission, for example, signature extended.	
	 Instance Type: The type of your DocuSign instance. This is an optional field. 	
	 Account ID: The API Account ID of the integration created on DocuSign. This is an optional field. 	
	For more details, see Create a DocuSign Adapter Connection in Using the DocuSign Adapter with Oracle Integration.	

Related Documentation

Using the DocuSign Adapter with Oracle Integration

Interact with Facebook

Use these recipes to retrieve the details of a Facebook page.



These recipes are available in the **REST** — **Facebook | Get Page Details** package in the Integration Store. Oracle provides these recipes as samples only. These recipes are meant only for guidance, and are not warranted to be error-free. No support is provided for these recipes.

Overview

Using the recipes in the REST — Facebook | Get Page Details package, you can:

- Get all the posts published by a Facebook page.
- Get the entire feed of a Facebook page, including posts and links published by the page or by the visitors to the page.

To run a Facebook recipe, you must configure the Oracle Facebook Connection with the details necessary to access Facebook. You can then trigger the recipe through a REST request and supply the page-id of the Facebook page for which you want to retrieve details. The Facebook Adapter in the recipe is subsequently invoked, and it fetches the details of the specified page from Facebook. The response containing the page data is returned to you.



Integration Name	Description	REST Commands and Example Payloads
Get Page Posts	Retrieves all the posts published by a Facebook page.	REST API Command:
		<pre>GET: https://<host:port>/ic/api/</host:port></pre>
		integration/v1/flows/rest/ORCL-R-
		FACEBO GET PAGE POSTS/1.0/getPosts?
		key=page-id&value={facebook page id}
		key page lawvalue (lacebook_page_la)
		Example Response Payload:
		{
		"postResource.definitions.postType":
		{
		"data": [
		{
		"created time":
		"2019-10-17T09:41:54+0000",
		"story": "",
		"id":
		"114257499511181_1240634186121094"
		}
]
		}
		}



Integration Name	Description	REST Commands and Example Payloads
	Retrieves the entire feed of a Facebook page, including posts	REST API Command:
	and links published by the page	<pre>GET: https://<host:port>/ic/api/</host:port></pre>
or by the visitors to the page.	integration/v1/flows/rest/ORCL-R-	
		<pre>FACEB_GET_PAGE_FEED/1.0/getPageDetails?</pre>
		<pre>key=page-id&value={facebook_page_id}</pre>
		Example Response Payload:
		{ "postResource.definitions.postType":
	{ "data": [{	
	"created_time":	
	"2021-03-12T04:13:57+0000",	
		"story": "James London wrote on
	ExamplePage's timeline - with David	

Ryan.",
"id":

{ "created time":

"Example message1", "id":

"Example message2", "id":

"Example message3", "id":

"Example message4", "id":

"Example message5", "id":

"Example message6", "id":

"QVFIUmdRR3VaSXJZAM2F1aEt

5RkNYems2ZAzR0R01tN2pYZ

9GemtibU1FcXdE",

"after":

"101637441997647 108967031264688" },

"101637441997647 108966757931382" },

"101637441997647 108965954598129" },

"101637441997647 108955377932520" },

"101637441997647 108577494636975" },

"101637441997647 108577171303674" },

"2021-03-12T04:13:38+0000", "message":

"2021-03-12T04:11:30+0000", "message":

"2021-03-12T03:55:31+0000", "message":

"2021-03-11T17:59:11+0000", "message":

"2021-03-11T17:58:26+0000", "message":

"2021-03-10T15:05:16+0000", "message":

"101637441997647_107388018089256" }], "paging": { "cursors": { "before":

PeXFNdy1NNk1weENIbG5IVkNaeldCRFo5aWd1Mng2R0JSZAElWVFBINzBnSmJBYzhTRjNfQ185VHZAvc3E

AWdGalZAQblpoeWNaOXN0Skd2WGlOMkZABMVNQZAl



Integration Name	Description	REST Commands and Example Payloads
		"QVFIUkwycGp3MG11YnVZAbFNBMzkwd09PanUtbzV KVHVPVT
		VuREwxcDdLNjkzNHRHR013ekl4RVhqbW9iNlJfOW5 QREp
		TQVVNVjFqZAGdheGNlMTdtOVowa2xSMHdNSkJQME5 3RzM
		4T2ZADQnhyUWpNMEY1Rk1XMDZAxbHd1ZAXpWLXdiU DNS" } } }

System and Access Requirements

- Oracle Integration
- Facebook

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Configure the Oracle REST Trigger Connection and the Oracle Facebook Connection present in the recipe package. You only need to configure these connections once. Both the recipes in the package use the same connections.

You must supply the following information when you configure these connections.

Connection Name	Parameters	
Oracle REST Trigger Connection	The connection parameters are already configured for you. Test and save the connection.	
Oracle Facebook Connection	Security Security Policy: FaceBook OAuth Authorization Code Credentials.	
	 Client Id: The App ID of your Facebook application. Client Secret: The App Secret of your Facebook application. Scope: The necessary scope parameters for the recipe. 	
	For more details, see Create a Facebook Adapter Connection in Using the Facebook Adapter with Oracle Integration.	

Related Documentation

Using the Facebook Adapter with Oracle Integration



Interact with LinkedIn

Use these recipes to get user profiles and to share content on LinkedIn.



These recipes are available in the Integration Store as:

- Oracle LinkedIn Get User Lite Profile
- Oracle LinkedIn Share
- Oracle LinkedIn Share User Content

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

Overview

The LinkedIn recipes use the REST Adapter and the LinkedIn Adapter to create an integration with a LinkedIn application. These recipes enable you to use basic LinkedIn services in an integration within Oracle Integration on behalf of an authenticated user. You can get the user's lite (basic) LinkedIn profile and share content on LinkedIn.

Recipe Name	Description	Parameters Required to Run Recipe
Get User Lite Profile	Retrieves the lite profile of the current LinkedIn user.	No parameter is required in the request message to run this recipe.
Share Message	Uses the LinkedIn Shares API to post a message to the account of the current LinkedIn user.	Provide the following sample payload in the request message:
	Note: LinkedIn still supports the Shares API, but plans to deprecate it in the near future. Also, the Share API doesn't support video.	<pre>{ "content": { "title": "Test Oracle LinkedIn Share message" }, "distribution": { "linkedInDistributionTarget": {} }, "owner": "<your_linkedin_account_urn>", "subject": "Test Oracle LinkedIn - Share 005", "text": { "text": "Test Oracle LinkedIn Share message" } }</your_linkedin_account_urn></pre>



Recipe Name	Description	Parameters Required to Run Recipe
Share User Generated Content	Uses the LinkedIn User Generated Content (UGC) Posts API to post a message to the account of the current LinkedIn user. Note: UGC Posts is LinkedIn's next generation of share experience encompassing video posts, share posts, and more. UGC Posts will eventually replace the Shares API.	<pre>Provide the following sample payload in the request message: { "author" : "<your_linkedin_account_urn>", "lifecycleState" : "PUBLISHED", "specificContent" : { "com.linkedin.ugc.ShareContent" : {</your_linkedin_account_urn></pre>

System and Access Requirements

Oracle Integration

Prerequisites

To use the LinkedIn recipes, you must satisfy the following prerequisites for creating a connection:

- Access the LinkedIn developer console, and then create and register an application.
 - When you register an application, LinkedIn provides a client ID and a client secret. The client ID and client secret are application-specific and are different for every application. You'll need these values when you configure the connection.
- Ensure that you have the credentials required to provide consent for access to the user's LinkedIn profile.
- Know the scope of the access request. Scopes enable you to specify which type of access you need. For example: r_emailaddress r_liteprofile w member social.

See Prerequisites for Creating a Connection in *Using the LinkedIn Adapter with Oracle Integration*.

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

To use the collection of LinkedIn recipes, you must configure the Oracle REST LinkedIn Connection and the Oracle LinkedIn Connection. You only need to configure these connections once. All LinkedIn recipes use the same two connections.

You must supply the following information when you configure these connections.

Connection Name	Parameters
Oracle REST LinkedIn Connection	Type: REST API Base URLConnection
	<pre>URL: https://oracle_integration_server name:443</pre>
	Security Policy: No security policy
Oracle LinkedIn Connection	The security policy grants you authorization access to the resources of the LinkedIn application. Configure the security for your LinkedIn connection by specifying the following parameters:
	Security Policy: Custom Security Policy
	 Client ID: Enter the client ID that you received when you registered your application with LinkedIn.
	 Client Secret: Enter the client secret that you received when you registered your application with LinkedIn.
	• Scope: Define the scope of the access request. Scopes enable you to specify which type of access you need, for example:
	r_emailaddress r_liteprofile w_member_social
	Click Provide Consent and enter the email address and password to authorize the application's access to the LinkedIn profile.
	See Prerequisites for Creating a Connection in <i>Using the LinkedIn Adapter with Oracle Integration</i>

Related Documentation

Using the LinkedIn Adapter with Oracle Integration

Interact with Slack

Use these recipes to create a channel for communication on the Slack application and post a message to a specific Slack channel, respectively.



These recipes are available in the Integration Store as:

- REST Slack | Create Channel
- REST Slack | Post Message

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



Overview

Using the Slack recipes, you can:

- Create a public or private channel for communication on the Slack application.
- Post a message to a public or private Slack channel.

To run the Slack recipes, you must configure the Oracle Slack Connection with the details necessary to access the Slack application. You can then trigger the recipes through a REST request and supply the necessary request payload.

- For the REST Slack | Create Channel recipe, you must supply the name and type for the new channel you want to create as the payload. The Slack Adapter in the recipe is subsequently invoked, and it creates the channel as specified in your Slack workspace. The response containing the channel-creation details is returned to you.
- For the REST Slack | Post Message recipe, you must supply the massage you want to
 post and the name of the channel you want to post the message in as payload. The Slack
 Adapter in the recipe is subsequently invoked, and it posts the message in the specified
 Slack channel. The response containing the message-post details is returned to you.



Integration Name	Description	REST Commands and Example Payloads
Create Channel	Creates a public or private channel for communication on the Slack application.	REST API Command:
		<pre>POST: https://<host:port>/ic/api/ integration/v1/flows/rest/ORCL-R- SLACK_CREATE_CHANNEL/1.0/createChannel</host:port></pre>
		Example Request Payload:
		ſ
		"channelName": "New-Channel", "is-private":"true"
		}
		Example Response Payload:
		{
		"ok": "true",
		"channel": {
		"id": "C01SGRKT84E",
		"name": "New-Channel",
		"is_channel": "true",
		"is_group": "false",
		"is_im": "false",
		"created": "1617017971",
		"is_archived": "false",
		"is_general": "false",
		"unlinked": "0",
		<pre>"name_normalized": "new-channel", "is shared": "false",</pre>
		"creator": "U01PUKGN6SK",
		"is ext shared": "false",
		"is org shared": "false",
		"is_pending_ext_shared": "false", "is member": "true",
		"is private": "true",
		"is_mpim": "false",
		"last read": "000000000.000000"
		}
		}



Integration Name	Description	REST Commands and Example Payloads
Push Notification	Posts a message to a public or private Slack channel.	REST API Command:
	•	<pre>POST: https://<host:port>/ic/api/ integration/v1/flows/rest/</host:port></pre>
		ORCL_R_SLACK_PUSH_NOTIFICA/1.0/ postMessage
		Example Request Payload:
		{
		"Message": "Test Message", "channelName": "new-channel"
		}
		Example Response Payload:
		{
		"ok": "true",
		"channel": "C01SGRKT84E", "ts": "1617031570.000300",
		"message": {
		"bot_id": "B01SC4EJ4JU",
		"type": "message",
		"text": "Test Message",
		"user": "",
		"ts": "1617031570.000300", "team": ""
		}
		}

System and Access Requirements

- Oracle Integration
- Slack

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Configure the Oracle REST Trigger Connection and the Oracle Slack Connection present in the recipe package. You only need to configure these connections once. Both the Slack recipes mentioned in this topic use the same connections.

You must supply the following information when you configure these connections.

Connection Name	Parameters
Oracle REST Trigger Connection	The connection parameters are already configured for you. Test and save the connection.



Connection Name	Parameters Security	
Oracle Slack Connection		
	 Client ID: The Client ID of the OAuth application created on Slack. 	
	 Client Secret: The Client Secret of the OAuth application created on Slack. 	
	 Scope: The list of user token scopes for which you want to grant permission. 	
	For more details, see Create a Slack Adapter Connection in <i>Using</i> the Slack Adapter with Oracle Integration.	

Related Documentation

Using the Slack Adapter with Oracle Integration

Interact with Twitter

Use these recipes to tweet and retweet messages, find followers and friends, get trending topics and locations, and look up status.



These recipes are available in the Integration Store as:

- Oracle Twitter Find All Follower IDs
- Oracle Twitter Find All Friend IDs
- Oracle Twitter Find All Twitter IDs Who Retweeted
- Oracle Twitter Find All Retweet My Tweets
- Oracle Twitter Get All Twitter Trending Locations
- Oracle Twitter Get All Twitter Trends Of Place
- Oracle Twitter Lookup Tweet Status
- Oracle Twitter Retweet
- Oracle Twitter Tweet

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

Overview

Twitter is an online social networking service that enables you to send and read short messages called **tweets**. When you sign in to your Twitter account, you can read and post tweets.

The Twitter recipes use the REST Adapter and the Twitter Adapter to create an integration with a Twitter application. The recipes let you interact with Twitter including tweeting messages, finding followers and friends, and getting information about trends.



Recipe Name	Description	Parameters Required to Run Recipe
Tweet Message	Posts a tweet (a status update) on behalf of the user for all followers and others to read on Twitter.	Provide the status parameter with the desired text to be posted in the request message.
Retweet Message	Reposts or forwards a tweet that was previously posted on Twitter by another user.	Provide the ID parameter with the tweet ID to be retweeted.
Get Retweeted Messages	Returns the most recent tweets authored by the authenticating user that have been retweeted by other users.	Provide any value to the dummy parameter in the request message.
Get Followers IDs	Returns a cursored collection of user IDs for every user who is following the specified user.	Provide the user_id and the screen_name of the Twitter account in the request message. Note that the screen_name is the Twitter
		ID without the @ sign.
Get Friends IDs	Returns a cursored collection of user IDs for every user who the specified user is following (otherwise known as their "friends").	Provide the user_id and the screen_name of the Twitter account in the request message. Note that the screen_name is the Twitter
		ID without the @ sign.
Get IDs of Who Retweeted	Returns a collection of up to 100 user IDs belonging to users who have retweeted the tweet specified by the ID parameter.	Provide the ID parameter with the ID of a tweet in the request message.
Get Trends for Place	Returns the top 50 trending topics for the location specified by the Where On Earth	Provide the ID parameter with the WOEID in the request message.
	Identifier (WOEID), if trending information is available for it.	For example, you can enter 2367105 to get the trending topics for Boston, 2459115 for New York, or 2442047 for Los Angeles. Also, you can use 1 to get global trending topics.
Get Trends Available	Returns the locations that Twitter has trending topic information for.	No parameter is required in the request message to run this recipe.
	The response is an array of "locations" that encode the location's WOEID and some other human-readable information such as a canonical name and country the location belongs in.	
Look Up Tweet Status	Returns fully-hydrated tweet objects for up to 100 tweets per request, as specified by the comma-separated values passed to the ID parameter.	Provide the following parameters in the request message: ID: Specifies the tweet ID.
	This recipe is useful to get the details of (hydrate) a collection of tweet IDs.	trim_user: When set to true (true/t/1), then include only the status authors numeric ID. (Optional).
		map: When used, tweets that don't exist or can't be viewed by the current user will still have their key represented but with an explicitly null value paired with it. (Optional).



System and Access Requirements

Oracle Integration

Prerequisites

To use the Twitter recipes, you must have a Twitter account and have access to the Twitter API for the integration. See Prerequisites for Creating a Connection in *Using the Twitter Adapter with Oracle Integration*.

Install, Configure, and Run the Recipes

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

To use the collection of Twitter recipes, you must configure the Oracle REST Twitter Connection and the Oracle Twitter Connection. You only need to configure these connections once. All Twitter recipes use the same two connections.

You must supply the following information when you configure these connections.

Connection Name	Parameters
Oracle REST Twitter Connection	Type: REST API Base URLConnection
	<pre>URL: https://oracle_integration_server name:443</pre>
	Security Policy: No security policy
Oracle Twitter Connection	The security policy grants you authorization access to the resources of the Twitter application. Configure the security for your Twitter connection by selecting the security policy and specifying the following parameters: Consumer Key Consumer Secret Access Token Access Secret You would have copied and saved the security values when you authorized your application to use your Twitter account. See Prerequisites for Creating a Connection in <i>Using the Twitter Adapter with Oracle Integration</i> .

Related Documentation

Using the Twitter Adapter with Oracle Integration



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.



This recipe is available as **SurveyMonkey** — **Slack | Post Notifications for Completed Surveys** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

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 package and configure connections and other resources within the package. Subsequently, you can activate and run the integration flow of the package 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 21.2.1.0.0 (210129.2200.39462) or higher
- SurveyMonkey
- Slack
- An account in SurveyMonkey with the Administrator role
- An account in Slack with the Administrator role

Topics:

- Before You Install the Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

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.

- 1. Configure SurveyMonkey
- 2. Configure Slack



Configure SurveyMonkey

You must perform the following prerequisite tasks in your SurveyMonkey instance.

- 1. Create and Configure a SurveyMonkey App
- 2. Create a Survey

Create and Configure a SurveyMonkey App

You have to create a SurveyMonkey app and configure it to access SurveyMonkey APIs from Oracle Integration.

- 1. Create 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.
 - a. 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.
 - b. Click Create App.
- 2. Click **SETTINGS** to open the app's settings.
- 3. Under the Settings section, enter the following in the OAuth Redirect URL field.

```
https://{OIC HOST}:{OIC SSL PORT}/icsapis/agent/oauth/callback
```

Replace $\tt OIC_HOST$ and $\tt OIC_SSL_PORT$ with specific values to your Oracle Integration instance.

- Get the SurveyMonkey API credentials.
 - Scroll 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.

- **5.** Select the scopes.
 - a. 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
 - b. Click Update Scopes.



You get a confirmation message that the scopes are successfully updated.

Create a Survey

You have to create a survey in SurveyMonkey for which you want to receive responses.

- 1. Log in to your SurveyMonkey account at https://www.surveymonkey.com/user/sign-in/.
- 2. On the SurveyMonkey Dashboard, click CREATE SURVEY on the title bar.
- 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/).

4. Preview your survey.

Configure Slack

You must perform the following prerequisite tasks in your Slack instance.

- 1. Create and Configure a Slack App
- 2. Set Up Your Slack WorkSpace and Channel

Create and Configure a Slack App

You have to create a Slack app, and configure it to receive notification messages sent by Oracle Integration.

- 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.
 - 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.

- 3. 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.
 - 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
 - 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.

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 who should receive the survey completed notification messages into the channel.

- 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, #survey-responses.
- **d.** Optionally, enter a suitable description in the **Description** field to let others know what the channel is about. For example: This channel notifies whenever an incident is created in ServiceNow.
- 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** $\stackrel{Q_+}{=}$ 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.

Install and Configure the Recipe

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

- 1. On the Oracle Integration Home page, scroll to the **Accelerators & Recipes** section.
- 2. Click Search All.

The list of available recipes appears.

3. Select the recipe you want to install, and click Install $\stackrel{+}{+}$.

A confirmation message informing the recipe was successfully installed appears, and the recipe now displays the **Installed** tag.

- 4. Click **Configure** to configure the following resources using the **Configuration**Editor:
 - Configure the Oracle SurveyMonkey Connection
 - Configure the Oracle Slack Connection
 - Configure Integration Schedule

Configure the Oracle SurveyMonkey Connection

You must configure connection details for your SurveyMonkey connection in Oracle Integration.

- 1. On the Configuration Editor page, select Oracle SurveyMonkey Connection.
- 2. Click Edit

The SurveyMonkey Connection window appears.

3. Under the Security section, in the **Client Id** and **Client Secret** fields, enter the client ID and secret you obtained earlier while creating and configuring a SurveyMonkey app.



See Create and Configure a SurveyMonkey App.

- 4. Click Provide Consent.
- 5. 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 SurveyMonkey appears.

- Enter your SurveyMonkey account credentials, then click Authorize and grant access.
- 7. On the SurveyMonkey Connection window, click Save.
- Click **Test** to ensure that your connection is successfully configured. In the dialog that displays, click **Test** again. You get a confirmation message if your test is successful.
- 9. Click **Back** to return to the Configuration Editor page.

Configure the Oracle Slack Connection

You must configure connection details for your Slack connection in Oracle Integration.

- 1. On the Configuration Editor page, select Oracle Slack Connection.
- 2. Click Edit

The Slack Connection window appears.

- 3. Under the Security section:
 - In the **Client ID** and **Client Secret** fields, enter the Client ID and Client Secret you obtained earlier while creating and configuring a Slack app.
 - In the Scope field, enter the permission scopes you configured while creating and configuring a Slack app.

See Create and Configure a Slack App.

4. Click Provide Consent.

A new browser window opens to approve access to Slack.

- 5. Click Allow.
- 6. On the Slack Connection window, click **Save**.
- Click Test to ensure that your connection is successfully configured. In the dialog that displays, click Test again. You get a confirmation message if your test is successful.
- 8. Click **Back** to return to the Configuration Editor page.

Configure Integration Schedule

Create a schedule for your integration to post Slack notification messages for completed SurveyMonkey survey responses at a date, time, and frequency of your choice.

1. On the Configuration Editor, select the Oracle SurveyMonkey Post Completed Responses integration.



- 2. Click **Actions** , then click **Add Schedule** from the menu. The page for defining the execution schedule is displayed.
- 3. Let's create a basic schedule, so leave the **Simple** radio button selected under the **Define recurrence** section.
- 4. In the **Frequency** field, select **Hours and Minutes**, and change the hours field to **6**. You've now created a schedule that runs the integration every six hours. Similarly, you can create a schedule that runs on particular days or weeks.
- 5. Additionally, you can specify the start and end dates for this schedule. In the **This** schedule is effective section, click the **From** field and select **Modify start date** to specify a date and time from when this schedule should run. Similarly, specify an expiry date and time in the **Until** field. In addition, you can also specify your preferred timezone.
- 6. Click **Save**, and then click **Back** twice to return to the Configuration Editor page. To create advanced schedules, see Define the Integration Schedule.

Activate and Run the Recipe

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

- 1. Activate the recipe.
 - a. On the Configuration Editor page, click Activate.
 - **b.** On the Activate Package dialog, click **Activate**. You get a confirmation message that the integration is activated.
- 2. Update the integration property values.
 - **a.** On the Configuration Editor page, select the Oracle SurveyMonkey Slack Post Completed Responses integration.
 - b. Click Actions (), and then click Update Property Values.
 - c. In the Update Property Values dialog, note the properties that have been configured ChannelName, SurveyName, and NotificationMail.
 - d. 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 know how to obtain the Slack channel ID, see Get the Slack Channel ID.
 - e. Click SurveyName and in the New Value field, enter the name of the survey that you had created in Slack.

See Create a Survey.

- f. 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.
- a. Click Submit.

You get a confirmation message 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.
- 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. Return back to Oracle Integration.
 - **b.** On the recipe's Configuration Editor page, select the Oracle SurveyMonkey Slack Post Completed Responses integration.
 - c. Click Run on the integration's row, and then select Submit Now.
 - In the resulting dialog, click Submit Now.
 You've successfully submitted the integration for a test run.
- 5. Monitor the execution of the integration flow.
 - **a.** In the Oracle Integration home page, click **Monitoring**, then **Integrations**, and then **Tracking**.
 - **b.** In the Track instances page, you can see the integration flow being triggered and executed successfully.
- 6. Check if a notification is posted in the Slack channel.
 - 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
- Using the Slack Adapter with Oracle Integration

Synchronize SurveyMonkey with Oracle NetSuite

Use this recipe to synchronize the contacts you add in SurveyMonkey to Oracle NetSuite.



This recipe is available as **Oracle SurveyMonkey NetSuite Customer Contact Sync** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.



Overview

This recipe creates an integration that synchronizes the contacts you add in SurveyMonkey group to a customer's contact list in Oracle NetSuite. The name of the group in SurveryMonkey must match the name of the customer in Oracle NetSuite.

You can add the contacts manually or import them from a CSV (comma separated values) file.

The integration that results from installing this recipe is named: Oracle SurveyMonkey NetSuite Customer Contact Sync.

System and Access Requirements

- Oracle Integration
- SurveyMonkey
- Oracle NetSuite

Install, Configure, and Run the Recipe

For more information and steps to install, configure, and run recipes, see Get Started with Integration Accelerators and Recipes.

Before you run the integration created with this recipe, you must configure the connections to SurveyMonkey and Oracle Netsuite.

To create a connection to SurveyMonkey, follow the steps described in Create a SurveyMonkey Adapter Connection in *Using the SurveyMonkey Adapter with Oracle Integration*.

To create a connection to Oracle NetSuite, follow the steps described in Create an Oracle NetSuite Adapter Connection in Using the Oracle NetSuite Adapter with Oracle Integration.

Related Documentation

To learn more about the SurveyMonkey and Oracle NetSuite adapters used in this recipe see:

- Using the SurveyMonkey Adapter with Oracle Integration
- Using the Oracle NetSuite Adapter with Oracle Integration

Other Recipes

Oracle is continually expanding its portfolio of recipes. In addition to the recipe documents listed here, you can find documentation for the latest recipes on our Recipes page in the Help Center.



B

Technical Accelerators Reference

In addition to prebuilt recipes, Oracle Integration includes a set of technical accelerators. Technical accelerators provide a common technical solution (for example, sending alerts on failures). They are meant to be called by another integration.

Topics:

- Alert Notification
- Resequence Messages

Alert Notification

Use the Alert Notification technical accelerator to send a message whenever an integration error or failure occurs. The notification can be sent by email, to a pager, in a JIRA ticket, or through a custom handler.

Topics:

- Description
- Key Parameters
- Supported Notification Types
- Connections
- Lookups
- Usage

Description

You can use the Alert Notification technical accelerator in all your integrations to customize the way in which you receive error notifications.

By adding this accelerator to all your integrations, you have a single consistent method for receiving notifications when an integration error or failure occurs. The turnaround time to resolve an issue is reduced, the issue gets directed to the right person, and maintenance is both streamlined and simplified.

You need to configure the REST connection (type, URL, and security policy). You'll also want to customize the values for the message ID, message subject, and notification type (for example, email or pager) for your organization,

You can't replace the connections or edit the integration in a technical accelerator developed by Oracle.

Key Parameters

The Alert Notification technical accelerator has several required fields or key parameters.

Field Name	Description	
instanceID	Instance ID of the calling service	
instanceID_string	Instance ID (for future use)	
IntgCode_Version	Integration code and version of the calling service	
messageID	A specific notification for an integration	
messageDetail	Information about the message corresponding to the message ID	
tracking_Var_1	Primary tracking variable of the calling service	
tracking_Var_2	Secondary tracking variable of the calling service	
tracking_Var_3	Third tracking variable of the calling service	
faultCode	Fault code of the calling service	
faultReason	Fault reason of the calling service	
faultDetails	Fault details of the calling service	

Supported Notification Types

The Alert Notification technical accelerator supports the following types of notifications:

- Email You provide a valid email address to send the error details to.
- PagerDuty You provide the ServiceID, EmailID, APIToken, and Assignee.
- JIRA Ticket You provide the JIRA user name, access token, project-key, assignee ID, and base URL.
- **Custom Error Handler** If you want your own error notification service, then you need to build a REST interface and provide the REST URL and credentials.

Connections

The Alert Notification technical accelerator uses two connections:

- Oracle Alerting Service Trigger This REST connection handles the trigger operation. The connection details are already configured. The connection is ready to be used.
- Oracle Alerting Service Invoke This REST connection handles the invoke operation. You must configure the following parameters:
 - Connection Type: REST API Base URL
 - Connection URL: Any valid URL
 - Security Policy: No Security Policy

Lookups

The Alert Notification technical accelerator uses three lookups. Edit the lookup values to customize the notification service for your organization.

The **ORCL-T-GENRIC_ENS_NOTIFICATION_SELECTOR_MSGID** lookup accepts these parameters:

Message_ID



- Message_Subject
- Notification Type

The **ORCL-T-GENRIC_ENS_NOTIFICATION_DATA** lookup stores the actual parameters for each notification type.

The **ORCL-T-GENRIC_ENS_NOTIFICATION_SELECTOR** accepts two parameters that define the notification type for the calling service.

- IntegrationCode_Version Specifies the integration code and version combination of the calling service.
- Notification_Type Specifies the kind of notification. For example, PAGERDUTY or Email.

Here are the values required for each notification type.

Notification Type	Parameter	Value To Be Provided	
Email	param1	Email of the user to whom the notification will be sent	
PagerDuty	param1 PagerDuty Admin Email ID		
	param2	PagerDuty API token	
	param3	PagerDuty Service ID	
	param4	PagerDuty Assignee	
JIRA	param1	JIRA Admin Username	
	param2	JIRA Access token	
	param3	JIRA Project Name	
	param4	JIRA Issue Type Name	
	param5	JIRA Assignee ID	
	param6	JIRA Base URL	
CUSTOM	param1	Custom values	
	param2		

Usage

Every integration should call the GenericErrorNotification Service from the Global Fault Handler.

You must configure the following parameters for the GenericErrorNotification Service:

- IntegrationCode_Version: For example, ORCL-BA-NS2SFDC_INVENITEM_CREATE::01.00.0000.
- InstanceID: Use the mapper function to get the instance ID of the calling service. For example, fn:format-number(nsmpr2:getFlowId(), "#").
- FaultCode: Map the errorCode from GlobalFaultObject.
- FaultReason: Map the reason from GlobalFaultObject.
- FaultDetails: Map the details from GlobalFaultObject.



Resequence Messages

The Resequence Messages technical accelerator provides a means for you to ensure that messages entering the system are processed in a specific order.



Oracle Integration comes with the Resequence Messages technical accelerator automatically installed. Therefore, this accelerator will already be listed as **INSTALLED** on the Accelerators and Recipes page when you sign in.

Topics:

- Description
- Prerequisite
- Key Parameters
- Connections
- Architecture
- Error Handling

Description

You often need to ensure that messages are processed in a strict order. Let's look at a pattern that enforces an order using Oracle Integration and the Resequence Messages technical accelerator. The pattern also deals with the case where you must limit the concurrency of calls to an endpoint system.

The Sequencing Problem

The basic problem is that you have a stream of requests that must be executed in order, for example, create account, update account address, and update account contacts. The latter two activities can't occur until the first one has completed.

Before you can sequence messages, you need to know the order in which the messages should be processed. So, you need to use some sort of sequencing ID. This ID could be a timestamp or an actual sequence identifier. If you're using timestamps, then the closer to the message origin that the timestamp is applied the better. For example, if you take the timestamp from when the message arrives in Oracle Integration, then a network delay may have already caused our messages to be out of order.

Typically, you don't want all messages to be in the same ordered sequence. In our account example, only messages for a given account need to be ordered. Messages for different accounts can execute in parallel. So, now you also need some sort of group ID to identify different sequence streams within your message stream.

After you have the messages and know their order, you can process them. Inherent in a resequencing solution is some sort of delay to allow messages to arrive out of order



and then be sorted into order. The size of the delay specifies how much time you can accept a message to be delayed before you go ahead without it.

The Resequencing Solution

The Resequence Messages technical accelerator includes a set of integrations, connections, and scripts that use standard Oracle Integration features. The integrations that handle the message resequencing are generic. You can use and reuse the technical accelerator to resequence different types of business integrations without modification to the integrations provided in the technical accelerator package.

The Resequence Messages technical accelerator:

- Processes the input message based on the desired sequence ID rather than on the order in which the messages arrive.
- Parks each message in storage for a certain period of time (parking time) so that any outof-sequence messages have a chance to be processed in the desired order.
- Lets you configure the maximum number of message groups being processed in parallel in order to throttle the outgoing calls.
- Takes care of all error handling, including system errors, network errors, and bad requests.

Prerequisite

The solution uses a database to store the input messages. You can create the required database tables by using a SQL script.

To create the required database tables:

- Search and download the DDL SQL script provided in this blog.
- Run the script to create the database tables.

Key Parameters

The Resequence Messages technical accelerator uses the following key parameters to reorder messages.

Field	Description	
gtype - Group Type	The type of stream. Different message types can be sequenced in parallel, for example, account updates and personnel updates are different group types.	
gid - Message Group	A field in the request that identifies a specific stream of messages to be sequenced.	
id - Message Identifier	A unique identifier for this message.	
sequenceld - Message Sequence	A field in the request or a timestamp that is used to determine how to sequence the messages in a stream.	
Parking Time	The amount of time that messages may be delayed in order to ensure messages are processed in the desired order.	
Message Concurrency	The maximum number of message groups to be processed in parallel.	



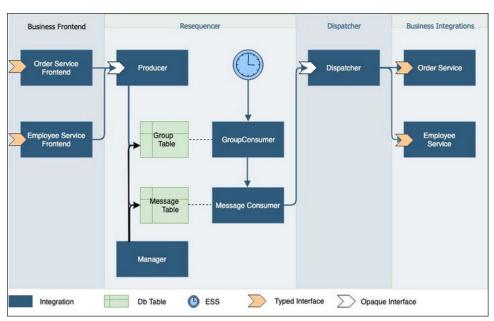
Connections

The Resequence Messages technical accelerator uses the following connections. After you install the accelerator, you need to configure each connection.

Connection	Туре	Description
RSQ DB	Invoke	ATP database used by the resequencer
RSQMessageConsumer	Trigger and invoke	Used to cause load distribution of calls to message consumer
RSQManager	Trigger	Used to invoke manager interface
RSQProducer	Trigger	Used to invoke producer interface
RSQDispatcher	Trigger	Used to invoke dispatcher interface
TestService	Invoke	Used to invoke a sample test service

Architecture

Let's look at the architecture for the Resequence Messages technical accelerator.



Business Front-end Integrations

An integration in this tier serves as the front end to the resequencer and the real business integration so that the message can go through the resequencer integrations to be reordered. The front-end integrations are specific to the use case and act as a means of converting typed requests into a common format, including group and sequencing fields.

Also, the front-end integrations receive the typed business payload, extract ID, and group ID from the message before calling the standardized producer message.



Resequencer Integrations

The Resequence Messages technical accelerator includes a set of resequencer integrations. These integrations handle the message resequencing and are generic. You can use them to resequence different types of business integrations. No additional modifications are necessary.

Resequencer Integrations	Description	
Producer	Serves as the entry point of the resequencer. The producer integration receives the resequencing message, creates a new row in the group table if it's not already there, and sets the status of the group to N . It then creates a message in the message table.	
	Sample message payload:	
	<pre>{ "gid": "gid", "gtype": "order", "id": "mid",</pre>	
Group Consumer	Detects the active groups and invokes the message consumer integration. The integration is scheduled to run every minute. When scheduling, use this expression:	
	<pre>FREQ=MINUTELY; INTERVAL=1;</pre>	
	The group consumer integration finds active groups, limiting parallelism to throttle outgoing calls to prevent overloading the target system. For each active group, the integration invokes a message consumer.	
Message Consumer	Processes active messages of the given group. It receives the group ID and type from the group consumer integration. It loads active messages of the group ordered by sequenceID. The messages have to be at least as old as the parking time. This ensures that there's a window for the message to arrive out of order but be processed in order.	
	The integration loops through active messages, marks the message status as \mathbf{P} , and invokes the dispatcher. Note that exceptions can occur here. After the dispatcher returns for a given message it is delete and the group status may be updated to mark the group status to be \mathbf{C} if there are no active messages, or \mathbf{N} if there are new active messages.	
	Sample message payload:	
	<pre>{ "id": "Engineering", "type": "employee" }</pre>	

Manager

The **manager** integration supervises the resequencer. It supports three operations.



Operation	Path and Method	Description
Get configs	Path: /configs Method: GET	Returns the config of all the types. Example of invocation: \$ curl https:// my.integration.cloud/ic/api/ integration/v1/flows/rest/ RSQMANAGER/1.0/configs -v -u username:password
Update config	Path: /configs/{type} Method: PUT	Update the config for the given type. Example of invocation: \$ curl -X PUT https:// my.integration.cloud/ic/api/ integration/v1/flows/rest/ RSQMANAGER/1.0/configs/employee -v -u username:password -H "Content-Type: application/ json" -d@config.json config.json example: { "maxConcurrent": 5, "timeWindow": 11 }
Recover Group	Path: /types/{type}/groups/ {group}/recover Method: PUT	Deletes stuck messages in the message table and reactivates the group by setting its status to 'N'. Example of invocation: \$ curl -X PUT https:// my.integration.cloud/ic/api/ integration/v1/flows/rest/ RSQMANAGER/1.0/types/employee/ groups/eng/recover -v -u username:password -H "Content- length: 0"

Dispatcher

The **dispatcher** is a request/response integration that reconstructs the original payload and sends it to the real backend integration. Unlike the resequencer integrations, the dispatcher isn't generic because it needs to invoke specific business integrations.

The dispatcher receives the message and converts the payload to the original typed business payload. It uses the group ID to find the business end point and invoke it synchronously. Exceptions can happen here.



Here is the dispatcher interface:

```
{ "id": "Engineering",

"gid" : "Zebra", "gtype" : "order", "sequenceId" : 123,

"payload" : "original payload"
}
```

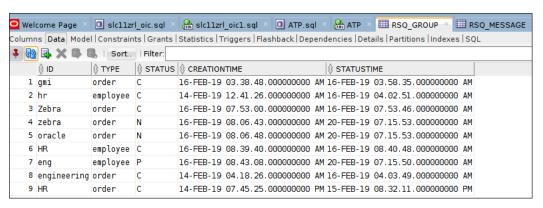
Business Integrations

A **business integration** is the real integration that processes the business messages. It has its own typed interface. For each business front-end integration, there should be a corresponding business integration.

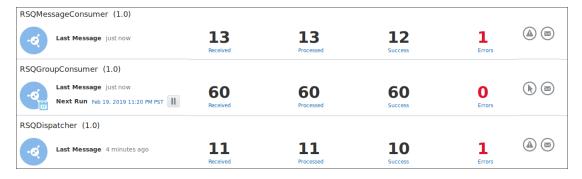
Error Handling

Exceptions can occur when the dispatcher integration invokes the business integration.

Exceptions bubble up to the message consumer integration and cause the message consumer instance to fail. When this happens, the group status stays at $\bf P$ in the database.



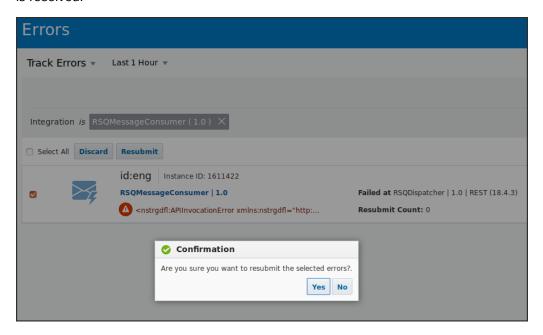
On the Monitoring Integration page, you can see the failed dispatcher instance and the message consumer instance.





Recover System Error

If the problem is caused by a system error, such as a networking issue, then you can recover by resubmitting the failed message consumer instance after the system error is resolved.



Recover Bad Request

If the error is caused by a bad request, then resubmitting the request will not help. In this case, you need to skip the bad request and move on. To do this, invoke the resequencer manager integration to remove the stuck message and reactivate the group:

\$ curl -X PUT https://my.integration.cloud/ic/api/
integration/v1/flows/rest/RSQMANAGER/1.0/types/employee/
groups/eng/recover -v -u username:password -H "Content-length:
0"

