

Oracle® Cloud

Using Recipes and Accelerators in Oracle Integration 3



G28326-14
February 2026



Oracle Cloud Using Recipes and Accelerators in Oracle Integration 3,

G28326-14

Copyright © 2025, 2026, 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, MySQL, and NetSuite 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

1 Get Started

Learn About Recipes and Accelerators	1
Find Recipes and Accelerators	2
Get More Details About Recipes and Accelerators	3

2 Install and Configure

Install a Recipe or Accelerator	1
Uninstall a Recipe or Accelerator	1
Configure Resources	2
Configure Resources in a Package	4

3 Activate and Monitor

Activate a Recipe or Accelerator	1
Monitor Integrations	1

4 Explore Recipes and Accelerators

Approve Invoices Using OCI AI Services	7
About This Recipe	7
System and Access Requirements	8
Before You Install the Recipe	8
Configure OCI Document Understanding	8
Configure OCI Generative AI	8
Configure OCI Generative AI Agents	8
Configure Oracle ERP Cloud	9
Install and Configure the Recipe	9
Configure the Oracle ERP Cloud Connection	9
Configure the REST Trigger Connection	9
Configure the Integration Flows	10
Configure the Lookup Table	11
Activate and Run the Recipe	11
Attach Files to Salesforce Records	12

About This Recipe	12
System and Access Requirements	12
Before You Install the Recipe	12
Install and Configure the Recipe	15
Configure the Oracle REST Trigger Connection	15
Configure the Oracle Salesforce Connection	15
Activate and Run the Recipe	16
Automate Customer Feedback Processing Using OCI AI Services and ServiceNow	17
About This Recipe	18
System and Access Requirements	18
Before You Install the Recipe	18
Install and Configure the Recipe	18
Configure the Microsoft Office 365 Outlook Connection	18
Configure the ServiceNow Connection	19
Configure the Integration Flow	19
Configure the Lookup Table	20
Activate and Run the Recipe	21
Calculate Total Balance	21
About This Recipe	22
System and Access Requirements	22
Before You Install the Recipe	22
Install and Configure the Recipe	22
Configure the Oracle SOAP Trigger Connection	22
Activate and Run the Recipe	22
Concatenate Messages	23
About This Recipe	23
System and Access Requirements	23
Install and Configure the Recipe	24
Configure the Oracle Rest Sample Trigger Connection	24
Activate and Run the Recipe	24
Copy List of Files on FTP Server	25
About This Recipe	25
System and Access Requirements	25
Before You Install the Recipe	25
Install and Configure the Recipe	26
Configure the Oracle FTP Connection	26
Activate and Run the Recipe	26
Copy Objects from OCI Object Storage to Microsoft SharePoint	27
About This Recipe	27
System and Access Requirements	27
Before You Install the Recipe	27
Configure OCI Object Storage	28

Configure SharePoint	28
Install and Configure the Recipe	29
Configure the SharePoint REST Connection	29
Configure the Lookup Table	30
Activate and Run the Recipe	30
Create Accounting Journals in Oracle Accounting Hub	31
About This Recipe	31
System and Access Requirements	31
Before You Install the Recipe	32
Access Your FTP Server	32
Configure Oracle ATP	32
Configure Oracle ERP Cloud	32
Configure Oracle Accounting Hub	33
Install and Configure the Recipe	33
Configure the Oracle ATP ImportAccountTransaction Connection	33
Configure the Oracle ERP SOAP ESS ImportAccountTrans Connection	34
Configure the Oracle FTP ImportAccountTransactions Connection	35
Configure the Oracle ERP ImportAccountTransactions Connection	35
Activate and Run the Recipe	36
Create Accounts in Oracle CX Sales and B2B Service for HubSpot Companies	37
About This Recipe	37
System and Access Requirements	37
Before You Install the Recipe	37
Configure HubSpot	37
Configure Oracle CX Sales and B2B Service	38
Install and Configure the Recipe	38
Configure the Oracle HubSpot Connection	38
Configure the Oracle B2B Connection	39
Configure the Lookup Table	39
Activate and Run the Recipe	40
Create Activities in Oracle Field Service Cloud for Oracle ERP Cloud Work Orders	41
About This Recipe	41
System and Access Requirements	41
Before You Install the Recipe	41
Configure Oracle Field Service Cloud	41
Configure Oracle ERP Cloud	42
Install and Configure the Recipe	43
Configure the Oracle Field Service Cloud Connection	43
Configure the Oracle ERP Cloud Connection	43
Configure the Lookup Table	44
Activate and Run the Recipe	44
Create Activities in Oracle Primavera Cloud for Jira Issues	45

About This Recipe	45
System and Access Requirements	45
Before You Install the Recipe	46
Configure Jira	46
Configure Oracle Primavera Cloud	46
Install and Configure the Recipe	46
Configure the Oracle Jira Connection	46
Configure the Oracle REST Primavera Connection	47
Configure the Lookup Table	47
Configure the Integration Flow	48
Activate and Run the Recipe	48
Create Advance Shipment Notices in Oracle Fusion Cloud SCM	49
About This Recipe	49
System and Access Requirements	49
Before You Install the Recipe	50
Configure Oracle ERP Cloud	50
Install and Configure the Recipe	50
Configure the ERP Connection	50
Activate and Run the Recipe	51
Create Alerts in Microsoft Teams for Jira Issues	52
About This Recipe	53
System and Access Requirements	53
Before You Install the Recipe	53
Configure Jira	53
Configure Microsoft Teams	54
Install and Configure the Recipe	55
Configure the Oracle Jira Connection	55
Configure the Oracle REST MS Teams Connection	55
Configure the Lookup Table	56
Activate and Run the Recipe	56
Create and Add a User to Multiple Groups in OCI	57
About This Recipe	57
System and Access Requirements	57
Before You Install the Recipe	57
Install and Configure the Recipe	57
Configure the OCI Domain REST Connection	58
Activate and Run the Recipe	58
Create Box Folders for ServiceNow Incidents	59
About This Recipe	59
System and Access Requirements	60
Recipe Schema	60
Before You Install the Recipe	60

Configure ServiceNow	60
Configure Box	62
Install and Configure the Recipe	63
Configure the Oracle ServiceNow Connection	64
Configure the Oracle Box Connection	64
Update Integration Properties	65
Activate and Run the Recipe	65
Create Box Folders for Zendesk Organizations	66
About This Recipe	66
System and Access Requirements	67
Before You Install the Recipe	67
Configure Zendesk	67
Configure Box	67
Install and Configure the Recipe	68
Configure the Oracle Zendesk Connection	68
Configure the Oracle Box Connection	69
Configure the Lookup Table	69
Activate and Run the Recipe	70
Create Customer Records in Oracle ERP Cloud	70
About This Recipe	70
System and Access Requirements	71
Assumptions	71
Before You Install the Recipe	71
Configure Oracle ERP Cloud	71
Install and Configure the Recipe	72
Configure the Oracle ERP Cloud Connection	72
Configure the Oracle REST Trigger Connection	72
Activate and Run the Recipe	72
Create Customer Records in Oracle NetSuite	74
About This Recipe	75
System and Access Requirements	75
Before You Install the Recipe	75
Install and Configure the Recipe	76
Configure the Oracle REST Connection	76
Configure the Oracle NetSuite Connection	76
Activate and Run the Recipe	77
Create Customers in Magento Commerce Cloud for Oracle ERP Cloud Accounts	78
About This Recipe	78
System and Access Requirements	79
Before You Install the Recipe	79
Configure Oracle ERP Cloud	79
Configure Magento Commerce Cloud	80

Install and Configure the Recipe	80
Configure the Oracle ERP Cloud Connection	80
Configure the Oracle Magento Commerce Cloud Connection	81
Update Integration Properties	81
Activate and Run the Recipe	82
Create Customers in Shopify for Salesforce Accounts	83
About This Recipe	83
System and Access Requirements	83
Before You Install the Recipe	83
Configure Salesforce	84
Configure Azure Service Bus	85
Configure Shopify	85
Install and Configure the Recipe	85
Configure the Oracle Salesforce Connection	85
Configure the Oracle Azure Service Bus Connection	86
Configure the Oracle Shopify Connection	86
Update Integration Properties	87
Activate and Run the Recipe	87
Create Invoices in Oracle ERP Cloud Using OCI Document Understanding AI Service	89
About This Recipe	89
System and Access Requirements	89
Before You Install the Recipe	90
Configure Microsoft Outlook	90
Configure Oracle ERP Cloud	90
Install and Configure the Recipe	90
Configure the Microsoft Outlook Connection	90
Configure the Oracle ERP Cloud Connection	91
Configure the REST Trigger Connection	91
Configure the Integration Flow	92
Configure the Lookup Table	92
Activate and Run the Recipe	92
Create Jira Issues for Exceptions in Integrations	93
About This Recipe	93
System and Access Requirements	93
Before You Install the Recipe	94
Create an API Token in Your Atlassian Account	94
Install and Configure the Recipe	94
Configure the Jira Connection	94
Activate and Run the Recipe	95
Create Jira Issues for Project Tasks in Oracle NetSuite	95
About This Recipe	96
System and Access Requirements	96

Before You Install the Recipe	96
Configure Jira	96
Configure Oracle NetSuite	97
Install and Configure the Recipe	98
Configure the Oracle Jira Connection	98
Configure the Oracle NetSuite Connection	99
Update Integration Properties	99
Activate and Run the Recipe	100
Create Jira Issues for ServiceNow Incidents	101
About This Recipe	101
System and Access Requirements	101
Before You Install the Recipe	101
Create an API Token in Your Atlassian Account	101
Configure ServiceNow	102
Install and Configure the Recipe	102
Configure the Oracle REST Jira Connection	102
Configure the Oracle ServiceNow Connection	102
Configure the Lookup Table	103
Activate and Run the Recipe	103
Create New User Records in Azure Active Directory	104
About This Recipe	104
System and Access Requirements	105
Before You Install the Recipe	105
Install and Configure the Recipe	105
Configure the Azure Active Directory Connection	105
Configure the Lookup Table	106
Activate and Run the Recipe	106
Create Oracle CPQ Transactions for Oracle Intelligent Advisor Interviews	107
About This Recipe	107
System and Access Requirements	108
Before You Install the Recipe	108
Configure Oracle Intelligent Advisor	108
Configure Oracle CPQ	108
Install and Configure the Recipe	108
Configure the Oracle Intelligent Advisor Connection	109
Configure the Oracle CPQ Connection	109
Configure the Lookup Tables	110
Update Integration Properties	110
Activate and Run the Recipe	110
Create Orders in Oracle Fusion Cloud Service for Oracle CPQ Opportunities	111
Create Products in Magento Commerce Cloud and QuickBooks for Oracle ERP Cloud Items	112
About This Recipe	112

System and Access Requirements	113
Before You Install the Recipe	113
Configure Oracle ERP Cloud	113
Configure GCP Pub Sub	113
Configure Magento Commerce Cloud	113
Configure QuickBooks	114
Install and Configure the Recipe	114
Configure the Oracle ERP Cloud Connection	114
Configure the GCP Pub Sub Invoke Connection	114
Configure the GCP Pub Sub Trigger Connection	115
Configure the Magento Commerce Cloud Connection	115
Configure the Lookup Tables	116
Activate and Run the Recipe	116
Create Products in Shopify for Items in Oracle ERP Cloud	118
About This Recipe	118
System and Access Requirements	118
Before You Install the Recipe	118
Configure Oracle ERP Cloud	118
Configure Shopify	119
Install and Configure the Recipe	119
Configure the Oracle ERP Cloud Connection	119
Configure the Shopify Connection	119
Activate and Run the Recipe	120
Create Project Tasks in Oracle NetSuite for Jira Issues	121
About This Recipe	121
System and Access Requirements	121
Before You Install the Recipe	121
Configure Jira	121
Configure Oracle NetSuite	122
Install and Configure the Recipe	123
Configure the Jira Connection	123
Configure the Oracle NetSuite Connection	124
Activate and Run the Recipe	124
Create Sales Orders in Oracle ERP Cloud for Salesforce Opportunities	125
About This Recipe	125
System and Access Requirements	125
Before You Install the Recipe	126
Configure Salesforce	126
Configure Oracle ERP Cloud	130
Install and Configure the Recipe	131
Configure the Salesforce Connection	131
Configure the Oracle ERP Cloud Connection	131

Configure the Lookup Table	132
Activate and Run the Recipe	132
Create Sales Orders in Oracle NetSuite for Invoices in Magento Commerce Cloud	133
About This Recipe	133
System and Access Requirements	133
Before You Install the Recipe	134
Configure Magento Commerce Cloud	134
Configure Oracle NetSuite	134
Install and Configure the Recipe	135
Configure the Magento Commerce Cloud Connection	135
Configure the Oracle NetSuite Connection	135
Activate and Run the Recipe	136
Create Sales Orders in Oracle NetSuite for Salesforce Opportunities	137
About This Recipe	137
System and Access Requirements	138
Assumptions	138
Before You Install the Recipe	138
Configure Salesforce	138
Configure Oracle NetSuite	143
Install and Configure the Recipe	144
Configure the Salesforce Connection	144
Configure the Oracle NetSuite Connection	145
Update Integration Properties	146
Activate and Run the Recipe	146
Create Sales Receipts in QuickBooks for Oracle NetSuite Invoices	147
About This Recipe	147
System and Access Requirements	147
Before You Install the Recipe	147
Configure Oracle NetSuite	147
Configure QuickBooks	149
Install and Configure the Recipe	149
Configure the Oracle NetSuite Connection	149
Configure the QuickBooks Connection	150
Configure the Lookup Table	150
Activate and Run the Recipe	151
Create Salesforce Accounts for SAP ERP Customers	151
About This Recipe	151
System and Access Requirements	152
Before You Install the Recipe	152
Configure SAP ERP	152
Configure Salesforce	152
Install and Configure the Recipe	154

Configure the SAP Connection	154
Configure the Salesforce Connection	155
Configure the Lookup Table	155
Activate and Run the Recipe	156
Create Salesforce Accounts for SAP S/4HANA Customers	157
About This Recipe	157
System and Access Requirements	157
Before You Install the Recipe	158
Configure SAP S/4HANA	158
Configure Salesforce	158
Install and Configure the Recipe	160
Configure the SAP S/4HANA Connection	160
Configure the Salesforce Connection	161
Configure the Lookup Table	161
Activate and Run the Recipe	162
Create Salesforce Cases for Jira Issues	162
About This Recipe	163
System and Access Requirements	163
Before You Install the Recipe	163
Create an API Token in Your Atlassian Account	163
Configure Salesforce	164
Install and Configure the Recipe	167
Configure the REST Jira Connection	167
Configure the Salesforce Connection	167
Activate and Run the Recipe	168
Create SAP ERP Orders for Closed Salesforce Opportunities	169
About This Recipe	169
System and Access Requirements	170
Before You Install the Recipe	170
Configure Salesforce	170
Configure SAP ERP	170
Install and Configure the Recipe	170
Configure the Salesforce REST Connection	170
Configure the SAP Connection	171
Update Integration Property	172
Configure the Lookup Table	172
Activate and Run the Recipe	172
Create Suppliers in SAP Ariba for Vendors in SAP ERP	173
About This Recipe	173
System and Access Requirements	173
Before You Install the Recipe	174
Configure SAP ERP	174

Configure SAP Ariba	174
Install and Configure the Recipe	174
Configure the SAP ERP Connection	174
Configure the Oracle SAP Ariba Connection	175
Configure the Lookup Table	175
Activate and Run the Recipe	175
Create Users in OCI Identity and Access Management for Persons in PeopleSoft	177
About This Recipe	177
System and Access Requirements	177
Before You Install the Recipe	177
Install and Configure the Recipe	183
Configure the IAM REST Connection	183
Configure the REST Trigger Connection	184
Activate and Run the Recipe	184
Create Workers in Oracle HCM Cloud	184
About This Recipe	185
System and Access Requirements	185
Before You Install the Recipe	185
Configure Oracle HCM Cloud	185
Access Your FTP Server and Create Folders	185
Install and Configure the Recipe	186
Configure the Oracle HCM Cloud	186
Configure the FTP Connection	186
Configure the Lookup Table	187
Update Integration Properties	187
Activate and Run the Recipe	188
Create Zendesk Tickets for Jira Issues	188
About This Recipe	188
System and Access Requirements	189
Before You Install the Recipe	189
Create an API Token in Your Atlassian Account	189
Configure Zendesk	190
Install and Configure the Recipe	190
Configure the REST Jira Connection	190
Configure the REST Zendesk Connection	190
Activate and Run the Recipe	191
Create Zendesk Tickets for Salesforce Cases	192
About This Recipe	192
System and Access Requirements	193
Before You Install the Recipe	193
Configure Salesforce	193
Configure Zendesk	197

Install and Configure the Recipe	197
Configure the Salesforce Connection	197
Configure the Zendesk Connection	198
Configure the Lookup Table	198
Activate and Run the Recipe	199
Delete Old Files	199
About This Recipe	199
System and Access Requirements	200
Before You Install the Recipe	200
Install and Configure the Recipe	200
Configure the FTP Connection	200
Configure the Lookup Table	201
Activate and Run the Recipe	201
Encrypt and Decrypt Data Using OCI Vault	202
About This Recipe	202
System and Access Requirements	202
Before You Install the Recipe	202
Install and Configure the Recipe	203
Configure the Oracle OCI Vault Connection	203
Configure the Lookup Table	204
Activate and Run the Recipe	204
Export Campaign Data in Bulk from Snowflake to Salesforce Marketing Cloud Using GCP Storage	205
About This Recipe	205
System and Access Requirements	206
Before You Install the Recipe	206
Configure Snowflake	206
Configure GCP Storage	206
Configure Salesforce Marketing Cloud	206
Install and Configure the Recipe	207
Configure the Snowflake Connection	207
Configure the GCP Storage Connection	207
Configure the Salesforce Marketing Cloud Connection	208
Configure the Lookup Table	208
Activate and Run the Recipe	209
Export Data in Bulk from Oracle ERP Cloud to an FTP Server	210
About This Recipe	210
System and Access Requirements	210
Before You Install the Recipe	211
Configure Oracle ERP Cloud	211
Access Your FTP Server	213
Install and Configure the Recipe	213

Configure the Oracle ERP Cloud Connection	213
Configure the Oracle SOAP ERP Integration Service Connection	214
Configure the Oracle FTP Connection	214
Configure the Lookup Table	215
Activate and Run the Recipe	215
Export Employee Images from Oracle HCM Cloud to an FTP Server	216
About This Recipe	216
System and Access Requirements	217
Before You Install the Recipe	217
Access Your FTP Server	217
Configure Oracle ERP Cloud	217
Configure Oracle Fusion SaaS	218
Install and Configure the Recipe	218
Configure the Oracle ERP Cloud Connection (Trigger, Invoke)	218
Configure the Oracle FTP Connection (Trigger, Invoke)	219
Configure the Lookup Table	219
Activate and Run the Recipe	220
Export Employee Records from Workday to Azure Active Directory	221
About This Recipe	221
System and Access Requirements	221
Before You Install the Recipe	222
Configure Workday	222
Configure Azure Active Directory	225
Install and Configure the Recipe	225
Configure the Workday Connection	225
Configure the Azure AD Connection	226
Configure the Lookup Tables	226
Activate and Run the Recipe	227
Export Employee Records from Workday to SAP	228
About This Recipe	228
System and Access Requirements	229
Before You Install the Recipe	229
Configure Workday	229
Configure SAP	243
Install and Configure the Recipe	245
Configure the Oracle Workday Connection	245
Configure the Oracle SAP Connection	245
Configure the Lookup Tables	246
Activate and Run the Recipe	247
Export Google Gmail Attachments to an FTP Server	248
About This Recipe	248
System and Access Requirements	248

Before You Install the Recipe	249
Configure Gmail	249
Access Your FTP Server and Create a Directory	250
Install and Configure the Recipe	250
Configure the Google Mail Connection	250
Configure the FTP Read Write Connection	251
Update Integration Properties	251
Activate and Run the Recipe	252
Export Invoices from PayPal to an FTP Server	252
About This Recipe	252
System and Access Requirements	252
Before You Install the Recipe	253
Configure PayPal	253
Access Your FTP Server	254
Install and Configure the Recipe	254
Configure the PayPal Connection	254
Configure the FTP Read Write Connection	254
Configure the REST PayPal Connection	255
Configure Lookups	255
Activate and Run the Recipe	256
Export Items in Bulk from Oracle ERP Cloud to Shopify	256
About This Recipe	256
System and Access Requirements	257
Recipe Schema	257
Assumptions	258
Before You Install the Recipe	258
Configure Oracle ERP Cloud	258
Configure Shopify	258
Access Your FTP Server	259
Install and Configure the Recipe	259
Configure the Oracle ERP Cloud Connection	259
Configure the Shopify REST Connection	260
Configure the REST Trigger Connection	260
Configure the FTP Connection	260
Configure the UCM Connection	261
Activate and Run the Recipe	261
Export Items with Variants from Oracle ERP Cloud to Shopify	262
About This Recipe	262
System and Access Requirements	262
Recipe Schema	262
Assumptions	264
Before You Install the Recipe	264

Configure Oracle ERP Cloud	264
Configure Shopify	264
Access Your FTP Server	265
Install and Configure the Recipe	265
Configure the Oracle ERP Cloud Connection	265
Configure the Shopify REST Connection	266
Configure the REST Trigger Connection	266
Configure the FTP Connection	266
Configure the UCM Connection	267
Configure the Lookup Tables	267
Activate and Run the Recipe	268
Export Messages from Apache Kafka to Amazon S3	269
About This Recipe	269
System and Access Requirements	269
Recipe Schema	269
Before You Install the Recipe	270
Configure Apache Kafka	270
Configure Amazon S3	271
Install and Configure the Recipe	272
Configure the Apache Kafka Connection	272
Configure the REST Amazon S3 Connection	273
Configure the Integration Flow	273
Activate and Run the Recipe	274
Export New Employee Data from Oracle HCM Cloud to an FTP Server	275
About This Recipe	275
System and Access Requirements	275
Before You Install the Recipe	276
Install and Configure the Recipe	276
Configure the Oracle HCM Cloud Connection	276
Configure the REST HCM Connection	276
Configure the FTP Connection	277
Activate and Run the Recipe	278
Export New Hire Information from Oracle HCM Cloud	278
About This Recipe	278
System and Access Requirements	279
Before You Install the Recipe	279
Configure Oracle HCM Cloud	279
Access Your FTP Server and Create a Directory	279
Install and Configure the Recipe	279
Configure the Oracle HCM Cloud Connection	279
Configure the FTP Read Write Connection	280
Activate and Run the Recipe	280

Export Oracle NetSuite Records to QuickBooks	281
About This Recipe	281
System and Access Requirements	282
Before You Install the Recipe	282
Configure Oracle NetSuite	282
Configure QuickBooks	283
Install and Configure the Recipe	283
Configure the Oracle NetSuite Connection	283
Configure the QuickBooks Connection	284
Activate and Run the Recipe	284
Export Purchase Orders from Oracle ERP Cloud to FTP Server	285
About This Recipe	285
System and Access Requirements	286
Before You Install the Recipe	286
Configure Oracle ERP Cloud	286
Access Your FTP Server	287
Install and Configure the Recipe	287
Configure the Oracle ERP Cloud Connection	287
Configure the FTP Connection	287
Configure the Lookup Table	288
Activate and Run the Recipe	288
Export Workday Employees to an FTP Server	289
About This Recipe	289
System and Access Requirements	289
Before You Install the Recipe	289
Configure Workday	290
Access Your FTP Server and Create a File	290
Install and Configure the Recipe	290
Configure the Oracle Workday Connection	290
Configure the FTP Connection	291
Configure the Integration Flow (Optional)	291
Activate and Run the Recipe	292
Extract Approved Expense Reports from SAP Concur and Write to an FTP Server	293
About This Recipe	293
System and Access Requirements	293
Before You Install the Recipe	294
Configure SAP Concur	294
Access Your FTP Server and Create a Directory	294
Install and Configure the Recipe	294
Configure the SAP Concur Connection	294
Configure the FTP Connection	295
Activate and Run the Recipe	295

Extract Data from Unstructured Files Using OCI Vision	296
About This Recipe	296
System and Access Requirements	296
Before You Install the Recipe	297
Install and Configure the Recipe	297
Configure the REST Vision API Connection	297
Configure the FTP Vision Invoice Rename Connection	298
Update Integration Properties	299
Activate and Run the Recipe	299
Extract Document Information using OCI AI Services	300
About This Recipe	300
System and Access Requirements	300
Before You Install the Recipe	300
Install and Configure the Recipe	300
Configure the OCI Document Understanding Connection	300
Activate and Run the Recipe	301
Fetch Data in Chunks from Oracle Autonomous Transaction Processing	302
About This Recipe	302
System and Access Requirements	302
Before You Install the Recipe	302
Install and Configure the Recipe	304
Configure the ATP Connection	304
Activate and Run the Recipe	305
Fusion HCM License and Certification Validation by EverCheck	305
About This Recipe	306
System and Access Requirements	306
Recipe Schema	306
Assumptions	308
Before You Install the Recipe	309
Configure Oracle HCM Cloud	309
Configure EverCheck	309
Install and Configure the Recipe	309
Configure the Oracle HCM Cloud Adapter Connection	310
Configure the FTP Connection	310
Configure the Fusion HCM Cloud REST Connection	311
Configure the Lookup Table	311
Configure the Oracle HCM Outbound Integration to EverCheck	312
Configure the Oracle HCM Inbound Integration from EverCheck	314
Bulk File Data Models	316
Activate and Run the Recipe	319
Fusion HCM License and Certification Validation by HireRight	320
About This Recipe	320

System and Access Requirements	320
Recipe Schema	321
Assumptions	325
Before You Install the Recipe	326
Configure Oracle HCM Cloud	326
Configure HireRight	326
Access an sFTP Server	327
Install and Configure the Recipe	327
Configure the Oracle HCM Cloud Connection	328
Configure the Oracle HCM Cloud REST Connection	328
Configure the HireRight REST Connection	329
Configure the sFTP Connection	329
Configure the OIC REST Connection	330
Configure the Lookup Table	330
Configure the Oracle HCM Initial Outbound Integration to HireRight	330
Configure the Oracle HCM Daily Outbound Integration to HireRight	332
Configure the Oracle HCM Recertification Integration to HireRight	335
Configure the Oracle HCM Inbound Integration from HireRight (HCM REST)	336
Configure the Oracle HCM Inbound Integration from HireRight (HCM Data Loader)	337
Activate and Run the Recipe	338
Generate an ANSI 834 Benefits File from Oracle HCM Cloud Extract File	339
About This Recipe	339
System and Access Requirements	340
Before You Install the Recipe	340
Install and Configure the Recipe	340
Configure the FTP Connection	340
Update Integration Properties	341
Activate and Run the Recipe	341
Generate and Email Patient Report Summary Using OCI AI Services	342
About This Recipe	342
System and Access Requirements	342
Before You Install the Recipe	343
Install and Configure the Recipe	343
Configure the Oracle Health Connection	343
Configure the REST Trigger Connection	343
Configure the Integration Flows	344
Configure the Lookup Table	344
Activate and Run the Recipe	345
Generate Oracle Hospitality Access Tokens	346
About This Recipe	346
System and Access Requirements	346
Assumptions	346

Install and Configure the Recipe	347
Configure the REST Hospitality Connection	347
Configure the Lookup Table	347
Activate and Run the Recipe	348
Get Details of a Facebook Page	348
About This Recipe	349
System and Access Requirements	349
Before You Install the Recipe	349
Install and Configure the Recipe	349
Configure the Facebook Connection	349
Configure the REST Connection	350
Activate and Run the Recipe	350
Get Incident Details From Oracle B2C Service	351
About This Recipe	351
System and Access Requirements	351
Before You Install the Recipe	351
Install and Configure the Recipe	352
Configure the Oracle B2C Service Connection	352
Configure the REST Trigger Connection	352
Activate and Run the Recipe	352
Get Opportunity Details from Oracle B2B Service	353
About This Recipe	353
System and Access Requirements	354
Before You Install the Recipe	354
Install and Configure the Recipe	354
Configure the Oracle CX Sales and B2B Service Connection	354
Configure the REST Connection	355
Activate and Run the Recipe	355
Get Organization Details from Oracle Service Cloud	356
About This Recipe	356
System and Access Requirements	356
Before You Install the Recipe	356
Install and Configure the Recipe	357
Configure the Oracle Engagement Cloud Instance Connection	357
Configure the Oracle RightNow ORG Connection	357
Configure the Oracle Service Cloud Connection	358
Configure the Oracle SOAP Get Organization Connection	358
Activate and Run the Recipe	358
Get Stocks Data	359
About This Recipe	359
System and Access Requirements	359
Before You Install the Recipe	360

Install and Configure the Recipe	360
Configure the REST Get Stock Quote Connection	360
Configure the REST Sample Trigger Connection	360
Activate and Run the Recipe	361
Get Weather Data	361
About This Recipe	362
System and Access Requirements	362
Before You Install the Recipe	362
Install and Configure the Recipe	362
Configure the REST Trigger Connection	362
Configure the REST Weather Connection	363
Activate and Run the Recipe	363
Handle Throttling Using Parking Lot Pattern	364
About This Recipe	364
System and Access Requirements	365
Recipe Schema	365
Before You Install the Recipe	366
Access Your FTP Server	366
Configure Oracle Autonomous Transaction Processing	366
Install and Configure the Recipe	366
Configure the Oracle FTP Connection	367
Configure the Oracle ATP Connection	367
Configure the Lookup Table	368
Activate and Run the Recipe	369
Import a File into Oracle Content Management	370
About This Recipe	370
System and Access Requirements	370
Before You Install the Recipe	371
Install and Configure the Recipe	371
Configure the Oracle REST Trigger Connection	371
Configure the Oracle REST Content Management Connection	371
Activate and Run the Recipe	372
Import a Payment Request into Oracle ERP Cloud	373
About This Recipe	373
System and Access Requirements	373
Recipe Schema	374
Payment Request File Definition	375
Before You Install the Recipe	379
Access Your FTP Server and Upload the Files	379
Configure Oracle ERP Cloud	380
Install and Configure the Recipe	380
Configure the FTP Connection	380

Configure the Oracle ERP Cloud Connection	381
Configure the Lookup Table	381
Activate and Run the Recipe	382
Import Absence Records from Oracle HCM Cloud to Google Calendar	383
About This Recipe	383
System and Access Requirements	383
Before You Install the Recipe	383
Configure Oracle HCM Cloud	383
Configure Google Calendar	384
Install and Configure the Recipe	384
Configure the Oracle HCM Cloud Connection	384
Configure the Oracle Google Calendar Connection	384
Configure the Lookup Table	385
Activate and Run the Recipe	385
Import Absence Records from Oracle HCM Cloud to Microsoft Calendar	387
About This Recipe	387
System and Access Requirements	387
Before You Install the Recipe	387
Configure Oracle HCM Cloud	387
Configure Microsoft Office 365 Calendar	388
Install and Configure the Recipe	388
Configure the Oracle HCM Cloud Connection	388
Configure the Oracle Microsoft Office 365 Calendar Connection	388
Activate and Run the Recipe	389
Import and Update Files in Bulk	390
About This Recipe	391
System and Access Requirements	391
Before You Install the Recipe	391
Install and Configure the Recipe	391
Configure the FTP Connection	391
Update Integration Properties	392
Activate and Run the Recipe	392
Import Candidate Records from an FTP Server to Taleo	393
About This Recipe	393
System and Access Requirements	393
Before You Install the Recipe	393
Access Your FTP Server and Upload the Files	393
Configure Taleo	394
Install and Configure the Recipe	394
Configure the FTP Import Candidates Connection	394
Configure the Taleo Import Candidates Connection	395
Configure the REST Import Candidates Connection	395

Activate and Run the Recipe	395
Import Contacts from Oracle Autonomous Transaction Processing to Oracle Responsys	397
About This Recipe	397
System and Access Requirements	397
Before You Install the Recipe	397
Install and Configure the Recipe	398
Configure the AQ Cloud Connection	398
Configure the Oracle Responsys Connection	399
Configure the Lookup Table	400
Update Integration Properties	401
Activate and Run the Recipe	401
Import Customer Records from an FTP Server to Oracle ERP Cloud	402
About This Recipe	402
System and Access Requirements	402
Before You Install the Recipe	403
Access Your FTP Server	403
Configure Oracle ERP Cloud	403
Install and Configure the Recipe	404
Configure the FTP Connection	404
Configure the Oracle ERP Cloud Connection	405
Activate and Run the Recipe	405
Import Customer Records from Shopify to Stripe	406
About This Recipe	407
System and Access Requirements	407
Before You Install the Recipe	407
Configure Shopify	407
Obtain Stripe API Keys	407
Install and Configure the Recipe	407
Configure the Shopify Connection	408
Configure the Stripe REST Trigger Connection	408
Activate and Run the Recipe	409
Import Customer Records in Bulk from Oracle JD Edwards to Salesforce	409
About This Recipe	409
System and Access Requirements	410
Before You Install the Recipe	410
Configure Oracle JD Edwards	410
Configure Salesforce	410
Install and Configure the Recipe	413
Configure the Oracle JD Edwards Connection	413
Configure the Salesforce Connection	413
Activate and Run the Recipe	414
Import Customers and Credit Card Transactions from SAP ASE (Sybase) to Oracle Unity	415

About This Recipe	415
System and Access Requirements	416
Assumption	416
Before You Install the Recipe	416
Configure SAP ASE (Sybase)	416
Configure Oracle Unity	417
Install and Configure the Recipe	417
Configure the Sybase Connection	417
Configure the Oracle Unity Connection	418
Activate and Run the Recipe	418
Import Documents from Oracle ERP Cloud to Adobe Sign	420
About This Recipe	420
System and Access Requirements	420
Before You Install the Recipe	420
Configure Oracle ERP Cloud	420
Configure Adobe Sign	421
Install and Configure the Recipe	421
Configure the Oracle ERP Cloud Connection	421
Configure the Adobe Sign Connection	422
Configure the Oracle ERP REST Connection	423
Configure the Lookup Table	423
Activate and Run the Recipe	423
Import Documents from Salesforce to Adobe Sign	425
About This Recipe	425
System and Access Requirements	425
Recipe Schema	425
Before You Install the Recipe	426
Configure Salesforce	426
Configure Adobe Sign	430
Install and Configure the Recipe	431
Configure the Salesforce Connection	431
Configure the Adobe REST Connection	432
Configure the REST Trigger Connection	433
Configure the Lookup Table	433
Activate and Run the Recipe	433
Import Employee Records from an FTP Server to Oracle HCM Cloud	434
About This Recipe	434
System and Access Requirements	434
Before You Install the Recipe	435
Access Your FTP Server	435
Configure Oracle HCM Cloud	435
Install and Configure the Recipe	436

Configure the FTP Connection	436
Configure the Oracle HCM Cloud Connection	436
Activate and Run the Recipe	437
Import Employee Records from Oracle HCM Cloud to Jira	438
About This Recipe	438
System and Access Requirements	438
Before You Install the Recipe	439
Configure Oracle HCM Cloud	439
Create an API Token in Your Atlassian Account	439
Install and Configure the Recipe	439
Configure the Oracle HCM Cloud Connection	439
Configure the REST Jira Connection	440
Configure the Lookup Table	440
Activate and Run the Recipe	440
Import Employee Records from Oracle HCM Cloud to Oracle E-Business Suite	441
About This Recipe	441
System and Access Requirements	441
Before You Install the Recipe	442
Configure Oracle HCM Cloud	442
Configure Oracle E-Business Suite	442
Install and Configure the Recipe	442
Configure the Oracle HCM Cloud Connection	442
Configure the Oracle E-Business Suite Connection	443
Update Integration Properties	443
Activate and Run the Recipe	444
Import Employee Records from Oracle HCM Cloud to Salesforce	445
About This Recipe	445
System and Access Requirements	445
Before You Install the Recipe	445
Configure Oracle HCM Cloud	445
Configure Salesforce	446
Install and Configure the Recipe	448
Configure the Oracle HCM Cloud Connection	448
Configure the Salesforce Connection	448
Configure the Lookup Table	449
Activate and Run the Recipe	449
Import Employee Records from Oracle HCM Cloud to Zendesk	450
About This Recipe	450
System and Access Requirements	451
Before You Install the Recipe	451
Configure Oracle HCM Cloud	451
Configure Zendesk	451

Install and Configure the Recipe	451
Configure the Oracle HCM Cloud Connection	451
Configure the Zendesk Connection	452
Configure the Lookup Table	452
Activate and Run the Recipe	453
Import Employee Vacation and Delegation Rules Report from Oracle HCM Cloud to Oracle Autonomous Transaction Processing	453
About This Recipe	454
System and Access Requirements	454
Before You Install the Recipe	454
Install and Configure the Recipe	455
Configure the ATP Connection	455
Configure the Oracle ERP Cloud Connection	456
Activate and Run the Recipe	456
Import Employees from SAP SuccessFactors to Oracle E-Business Suite	457
About This Recipe	457
System and Access Requirements	457
Before You Install the Recipe	458
Configure SAP SuccessFactors	458
Configure Oracle E-Business Suite	458
Install and Configure the Recipe	458
Configure the SAP SuccessFactors Connection	458
Configure the Oracle E-Business Suite Connection	460
Configure the Lookup Table	461
Activate and Run the Recipe	461
Import Files from an FTP Server to Oracle EPM Cloud	462
About This Recipe	462
System and Access Requirements	463
Before You Install the Recipe	463
Install and Configure the Recipe	463
Configure the FTP Connection	463
Configure the REST EPM Connection	464
Update Integration Properties	464
Activate and Run the Recipe	464
Import Files from an FTP Server to Oracle NetSuite	465
About This Recipe	465
System and Access Requirements	465
Before You Install the Recipe	466
Access Your FTP Server and Upload the Files	466
Configure Oracle NetSuite	466
Install and Configure the Recipe	467
Configure the FTP Read-Write Connection	467

Configure the Oracle NetSuite Connection	467
Configure the Integration Flow	468
Activate and Run the Recipe	469
Import Files from File Server to OCI Object Storage	470
About This Recipe	470
System and Access Requirements	470
Before You Install the Recipe	470
Configure File Server	470
Configure OCI Object Storage	471
Install and Configure the Recipe	471
Configure the Lookup Table	471
Activate and Run the Recipe	471
Import Financial Journal Entries from Amazon S3 to Oracle ERP Cloud	472
About This Recipe	472
System and Access Requirements	473
Before You Install the Recipe	473
Configure Amazon S3	473
Configure Oracle ERP Cloud	474
Install and Configure the Recipe	475
Configure the Oracle Amazon S3 Connection	475
Configure the Oracle ERP Cloud Connection	475
Configure the Lookup Table	476
Update Integration Properties	476
Activate and Run the Recipe	476
Import Financial Journal Entries from an FTP server to Oracle ERP Cloud	477
About This Recipe	477
System and Access Requirements	478
Before You Install the Recipe	478
Access Your FTP Server	478
Configure Oracle ERP Cloud	478
Install and Configure the Recipe	479
Configure the Oracle FTP Connection	479
Configure Oracle ERP Cloud	480
Configure the Lookup Table	480
Update Integration Properties	481
Activate and Run the Recipe	481
Import Leads from SugarCRM to Oracle ERP Cloud	482
About This Recipe	482
System and Access Requirements	482
Before You Install the Recipe	482
Configure SugarCRM	482
Configure Oracle ERP Cloud	483

Install and Configure the Recipe	483
Configure the Oracle SugarCRM Connection	483
Configure the Oracle ERP Cloud Connection	484
Activate and Run the Recipe	484
Import Marketo Leads to Oracle Autonomous Data Warehouse	485
About This Recipe	485
System and Access Requirements	485
Before You Install the Recipe	486
Configure Marketo	486
Configure Oracle Autonomous Data Warehouse	486
Install and Configure the Recipe	487
Configure the Oracle Marketo Connection	487
Configure the Oracle ADW Connection	487
Activate and Run the Recipe	488
Import Offer Letters from Oracle HCM Cloud to Adobe Sign	488
About This Recipe	489
System and Access Requirements	489
Recipe Schema	489
Before You Install the Recipe	490
Configure Oracle HCM Cloud	490
Configure Adobe Sign	490
Configure Adobe PDF Service Authentication	491
Install and Configure the Recipe	491
Configure the Oracle Adobe eSign REST Connection	491
Configure the Oracle REST Trigger Connection	493
Configure the Oracle FTP Connection	493
Configure the Oracle HCM Cloud Connection	493
Configure the Oracle Adobe IDM Connection	494
Configure the Oracle Adobe REST Connection	494
Configure the Lookup Table	495
Activate and Run the Recipe	495
Import Oracle Commerce Cloud Orders to Oracle Intelligent Advisor Interviews	496
About This Recipe	496
System and Access Requirements	497
Before You Install the Recipe	497
Configure Oracle Commerce Cloud	497
Configure Oracle Intelligent Advisor	497
Install and Configure the Recipe	497
Configure the Oracle Commerce Cloud Connection	497
Configure the Oracle Intelligent Advisor Connection	498
Update Integration Property	498
Activate and Run the Recipe	498

Import Orders from Oracle ERP Cloud to Oracle Autonomous Data Warehouse	499
About This Recipe	499
System and Access Requirements	499
Recipe Schema	500
Before You Install the Recipe	500
Configure Oracle ERP Cloud	500
Configure Apache Kafka	501
Configure Oracle Autonomous Data Warehouse	502
Install and Configure the Recipe	502
Configure the Oracle ERP Cloud Connection	502
Configure the Apache Kafka Connection	503
Configure the Oracle ADW Connection	503
Activate and Run the Recipe	504
Import Orders from Shopify to Oracle ERP Cloud	504
About This Recipe	505
System and Access Requirements	505
Before You Install the Recipe	505
Configure Shopify	505
Configure Oracle ERP Cloud	505
Install and Configure the Recipe	506
Configure the Oracle Shopify Connection	506
Configure the Oracle ERP Cloud Connection	507
Update Integration Property	508
Configure the Lookup Tables	508
Configure the Integration Flow (Optional)	508
Activate and Run the Recipe	509
Import Orders from Shopify to Salesforce	510
About This Recipe	510
System and Access Requirements	510
Before You Install the Recipe	511
Configure Shopify	511
Configure Salesforce	511
Install and Configure the Recipe	514
Configure the Oracle Shopify Connection	515
Configure the Oracle Salesforce Connection	515
Update Integration Property	516
Configure the Lookup Table	516
Activate and Run the Recipe	516
Import Orders from Zuora to a Hasura Supported Database	517
About This Recipe	517
System and Access Requirements	518
Before You Install the Recipe	518

Configure Zuora	518
Configure Hasura	519
Install and Configure the Recipe	519
Configure the Oracle Zuora Connection	519
Configure the Oracle GraphQL Hasura Connection	520
Update Integration Properties	520
Activate and Run the Recipe	521
Import Products from Salesforce to Shopify	522
About This Recipe	522
System and Access Requirements	522
Before You Install the Recipe	522
Configure Salesforce	523
Configure Shopify	523
Install and Configure the Recipe	523
Configure the Salesforce REST Connection	524
Configure the GraphQL Shopify Connection	524
Configure the Lookup Table	525
Update Integration Properties	525
Activate and Run the Recipe	525
Import Purchase Orders from Oracle ERP Cloud to Microsoft SQL Server	526
About This Recipe	526
System and Access Requirements	526
Before You Install the Recipe	526
Configure Oracle ERP Cloud	527
Configure the Microsoft SQL Server Database	527
Install and Configure the Recipe	528
Configure the Oracle ERP Cloud Connection	529
Configure the Microsoft SQL Server Database	529
Activate and Run the Recipe	531
Import Purchase Orders from Oracle JD Edwards EnterpriseOne to Oracle ERP Cloud	531
About This Recipe	531
System and Access Requirements	532
Before You Install the Recipe	532
Configure Oracle JD Edwards EnterpriseOne	532
Configure Oracle ERP Cloud	533
Install and Configure the Recipe	533
Configure the Oracle JD Edwards EnterpriseOne Connection	533
Configure the Oracle ERP Cloud Connection	534
Configure the Lookup Tables	534
Activate and Run the Recipe	535
Import Sales Orders from Oracle ERP Cloud to MySQL	536
About This Recipe	536

System and Access Requirements	536
Before You Install the Recipe	536
Configure Oracle ERP Cloud	536
Configure the MySQL Database	537
Install and Configure the Recipe	538
Configure the Oracle ERP Cloud Connection	538
Configure the MySQL Connection	539
Update Integration Properties	539
Activate and Run the Recipe	540
Import Sales Orders from Oracle NetSuite to Oracle Autonomous Data Warehouse	540
About This Recipe	540
System and Access Requirements	541
Before You Install the Recipe	541
Configure Oracle NetSuite	541
Configure Oracle Autonomous Data Warehouse	542
Install and Configure the Recipe	543
Configure the Oracle NetSuite Connection	543
Configure the Oracle ADW Connection	544
Activate and Run the Recipe	544
Import Segments from Oracle Unity to Salesforce Marketing Cloud	545
About This Recipe	545
System and Access Requirements	546
Assumptions	546
Before You Install the Recipe	546
Configure Oracle Unity	546
Configure Salesforce Marketing Cloud	547
Install and Configure the Recipe	549
Configure the REST Oracle Unity Connection	549
Configure the FTP Salesforce Marketing Cloud Connection	550
Configure the SOAP Salesforce Marketing Cloud Connection	550
Configure the Lookup Tables	551
Activate and Run the Recipe	552
Import Supplier Records Asynchronously from Oracle ERP Cloud to Oracle Autonomous Transaction Processing	553
About This Recipe	553
System and Access Requirements	554
Before You Install the Recipe	554
Configure Oracle ERP Cloud	554
Configure Oracle Autonomous Transaction Processing	554
Install and Configure the Recipe	557
Configure the Oracle ERP Cloud Connection	558
Configure the Oracle Autonomous Transaction Processing Connection	558

Activate and Run the Recipe	559
Import Supplier Records from SAP S/4HANA to Oracle ERP Cloud	559
About This Recipe	559
System and Access Requirements	560
Before You Install the Recipe	560
Configure SAP S/4HANA	560
Configure Oracle ERP Cloud	560
Install and Configure the Recipe	560
Configure the SAP S/4HANA Connection	560
Configure the Oracle ERP Cloud Connection	561
Configure the Lookup Table	561
Activate and Run the Recipe	562
Import Suppliers from File Server to Oracle ERP Cloud	562
About This Recipe	563
System and Access Requirements	563
Before You Install the Recipe	563
Configure File Server	563
Configure Oracle ERP Cloud	564
Install and Configure the Recipe	564
Configure the Oracle ERP Cloud Connection	564
Configure the Lookup Table	564
Activate and Run the Recipe	565
Import Users from Oracle HCM Cloud to Okta	565
About This Recipe	566
System and Access Requirements	566
Before You Install the Recipe	566
Configure Oracle HCM Cloud	566
Configure Okta API Key	566
Install and Configure the Recipe	566
Configure the Oracle HCM Cloud Connection	567
Configure the REST Okta Connection	567
Activate and Run the Recipe	568
Import Worker Data from an FTP Server to Oracle HCM Cloud	568
About This Recipe	568
System and Access Requirements	569
Before You Install the Recipe	569
Access Your FTP Server	569
Configure Oracle HCM Cloud	569
Install and Configure the Recipe	570
Configure the FTP Connection	570
Configure the Oracle HCM Cloud Connection	570
Activate and Run the Recipe	571

Increment Variable in a Loop	572
About This Recipe	572
System and Access Requirements	572
Install and Configure the Recipe	572
Configure the REST Connection	572
Activate and Run the Recipe	573
Integrate DiCentral with Oracle SCM Cloud	574
About This Recipe	574
System and Access Requirements	575
Assumptions	575
Before You Install the Recipe	575
Configure Oracle SCM Cloud	575
Configure DiCentral	576
Install and Configure the Recipe	577
Configure the REST OIC API Connection	577
Configure the Oracle ERP Cloud DiCentral Connection	577
Configure the REST DiCentral Sales Order Connection	578
Configure the REST DiCentral Shipment Invoice Connection	578
Configure the REST OIC API Connection	579
Configure the Lookup Table	579
Activate and Run the Recipe	582
Integrate KanbanBOX with Oracle SCM Cloud	585
About This Recipe	586
System and Access Requirements	587
Assumptions	587
Before You Install the Recipe	587
Configure Oracle SCM Cloud	587
Configure KanbanBOX	588
Install and Configure the Recipe	589
Configure the REST OIC API Connection	589
Configure the REST KanbanBOX Connection	589
Configure the ERP Cloud Connection KanbanBOX	590
Configure the Lookup Tables	590
Activate and Run the Recipe	594
Example Request Payloads	596
Integrate Oracle Fusion Cloud IoT Intelligent Applications with Oracle Field Service Cloud	598
Integrate Oracle MICROS Reporting and Analytics Sales Data from ATP with Oracle Financials Accounting Hub	599
About This Recipe	599
System and Access Requirements	600
Before You Install the Recipe	600
Configure Oracle ERP Cloud	600

Configure Oracle Autonomous Transaction Processing	602
Install and Configure the Recipe	602
Configure the Oracle ERP Cloud MICROS Connection	602
Configure the Oracle ATP MICROS Connection	602
Configure the Lookup Tables	603
Activate and Run the Recipe	605
Integrate Oracle MICROS Reporting and Analytics with Oracle Autonomous Transaction Processing	606
About This Recipe	606
System and Access Requirements	607
Before You Install the Recipe	607
Configure Oracle MICROS Reporting and Analytics	608
Configure Oracle Autonomous Transaction Processing	608
Install and Configure the Recipe	609
Configure the REST OIC API Connection	609
Configure the Oracle MICROS Reporting and Analytics BI API Connection	610
Configure the ATP Connection	611
Configure the Lookup Table	611
Activate and Run the Recipe	612
Integrate with DocuSign	612
About the Recipes	613
System and Access Requirements	613
Before You Install the Recipes	613
Configure DocuSign	614
Install and Configure the Recipes	614
Configure the DocuSign Connection	615
Configure the REST Connection	615
Activate and Run the Recipes	616
Interact with LinkedIn	616
About The Recipes	617
System and Access Requirements	619
Before You Install the Recipes	620
Install and Configure the Recipes	620
Configure the REST LinkedIn Connection	620
Configure the LinkedIn Connection	620
Activate and Run the Recipes	621
Interact with Slack	621
About the Recipes	622
System and Access Requirements	624
Before You Install the Recipes	625
Install and Configure the Recipes	626
Configure the Slack Connection	626

Configure the REST Trigger Connection	626
Activate and Run the Recipes	627
Interact with Twitter	627
About The Recipes	628
System and Access Requirements	629
Before You Install the Recipes	629
Install and Configure the Recipes	629
Configure the REST Twitter Connection	629
Configure the Twitter Connection	630
Activate and Run the Recipes	630
Invoke a Local Integration	631
About This Recipe	631
System and Access Requirements	631
Install and Configure the Recipe	632
Configure the REST Sample Trigger Connection	632
Configure the REST Sample Endpoint Interface Connection	632
Activate and Run the Recipe	632
Invoke a Non-Local Integration	633
About This Recipe	633
System and Access Requirements	633
Before You Install the Recipe	634
Install and Configure the Recipe	634
Configure the REST Sample Hello World Invoke Connection	634
Configure the Oracle REST Trigger Connection	634
Activate and Run the Recipe	635
Invoke Pop-Up Notifications for Events in Oracle Fusion Applications	635
About This Recipe	636
System and Access Requirements	636
Before You Install the Recipe	636
Install and Configure the Recipe	636
Configure the REST OIC API Connection	637
Configure the Oracle ERP Cloud Notification Connection	637
Activate and Run the Recipe	638
Log a Request Message	639
About This Recipe	639
System and Access Requirements	639
Install and Configure the Recipe	639
Configure the REST Trigger Connection	639
Activate and Run the Recipe	640
Manage Objects in OCI Object Storage	640
About This Recipe	640
System and Access Requirements	641

Before You Install the Recipe	641
Install and Configure the Recipe	641
Configure the Lookup Table	641
Activate and Run the Recipe	642
Manage ServiceNow Incidents	642
About This Recipe	642
System and Access Requirements	645
Install and Configure the Recipe	645
Configure the Oracle ServiceNow Connection	645
Configure the Oracle REST Connection	646
Activate and Run the Recipe	646
Migrate Integration Artifacts Between Instances	647
About This Recipe	647
System and Access Requirements	647
Install and Configure the Recipe	647
Configure the Oracle REST Connection (Trigger)	648
Configure the Oracle REST Source Connection (Invoke)	648
Configure the Oracle REST Target Connection (Invoke)	648
Configure the Lookup Table	649
Activate and Run the Recipe	649
Perform Arithmetic Operations	650
About This Recipe	650
System and Access Requirements	661
Before You Install the Recipe	661
Install and Configure the Recipe	661
Configure the Oracle SOAP Trigger Connection	661
Activate and Run the Recipe	662
Perform Multiple Operations on Resources	662
About This Recipe	662
System and Access Requirements	663
Install and Configure the Recipe	663
Configure the Oracle REST Trigger Connection	663
Activate and Run the Recipe	663
Process Inventory, Order, and Shipping Info Between Oracle ERP/SCM Cloud and Oracle WMS Cloud	664
Post Slack Notifications for Completed SurveyMonkey Surveys	665
About This Recipe	665
System and Access Requirements	666
Before You Install the Recipe	666
Configure SurveyMonkey	666
Configure Slack	667
Install and Configure the Recipe	669

Configure the Oracle SurveyMonkey Connection	669
Configure the Oracle Slack Connection	670
Activate and Run the Recipe	670
Post Slack Notifications for New Marketo Leads	671
About This Recipe	672
System and Access Requirements	672
Recipe Schema	672
Before You Install the Recipe	673
Configure Marketo	673
Configure Slack	674
Install and Configure the Recipe	675
Configure the Oracle Marketo Connection	676
Configure the Oracle Slack Connection	676
Update Integration Properties	676
Activate and Run the Recipe	677
Post Slack Notifications for Oracle CPQ Quote Approvals	678
About This Recipe	678
System and Access Requirements	678
Before You Install the Recipe	678
Configure Oracle CPQ	679
Configure Slack	680
Install and Configure the Recipe	682
Configure the Oracle CPQ Connection	682
Configure the Oracle Slack Connection	682
Update Integration Property	683
Activate and Run the Recipe	683
Post Slack Notifications for Oracle ERP Cloud Purchase Order Approvals	684
About This Recipe	684
System and Access Requirements	684
Before You Install the Recipe	685
Configure Oracle ERP Cloud	685
Configure Slack	685
Install and Configure the Recipe	685
Configure the Oracle ERP Cloud Connection	685
Configure the Slack Connection	686
Update Integration Properties	686
Activate and Run the Recipe	687
Post Slack Notifications for ServiceNow Activities	687
About This Recipe	687
System and Access Requirements	688
Before You Install the Recipe	688
Configure ServiceNow	688

Configure Slack	688
Install and Configure the Recipe	691
Configure the ServiceNow Connection	691
Configure the Slack Connection	691
Update Integration Properties	692
Activate and Run the Recipe	692
Post Slack Notifications for Stripe Charges	694
About This Recipe	694
System and Access Requirements	694
Before You Install the Recipe	695
Configure Stripe	695
Configure Slack	695
Install and Configure the Recipe	696
Configure the Stripe REST Trigger Connection	696
Configure the Slack Connection	697
Activate and Run the Recipe	697
Publish and Subscribe to Oracle Cloud Infrastructure Streaming Service	698
About This Recipe	698
System and Access Requirements	699
Before You Install the Recipe	699
Install and Configure the Recipe	699
Configure the Oracle Cloud Infrastructure Streaming Connection	699
Configure the Integration Flow	700
Activate and Run the Recipe	700
Publish Shopify Orders to GCP Pub Sub	701
About This Recipe	701
System and Access Requirements	701
Before You Install the Recipe	702
Configure Shopify	702
Configure GCP Pub Sub	702
Install and Configure the Recipe	702
Configure the Shopify Connection	702
Configure the GCP Pub Sub Connection	703
Activate and Run the Recipe	703
Read and Write Files on an FTP Server	704
About This Recipe	704
System and Access Requirements	704
Before You Install the Recipe	704
Install and Configure the Recipe	705
Configure the FTP Connection	705
Activate and Run the Recipe	705
Resequence and Process Messages in a Specific Order	706

About This Recipe	706
System and Access Requirements	708
Before You Install the Recipe	708
Install and Configure the Recipe	711
Configure the ATP DB RSQ Connection	711
Configure the REST Trigger RSQManager Connection	711
Configure the REST Trigger EmployeeService Connection	712
Configure the REST RSQMessageConsumer Connection	712
Configure the REST Employee Consumer Connection	712
Configure the REST RSQProducer Connection	713
Configure the Oracle REST OrderService Connection	714
Configure the Oracle REST RSQDispatcher Connection	714
Activate and Run the Recipe	715
Send a Hello Message	716
About This Recipe	716
System and Access Requirements	716
Install and Configure the Recipe	716
Configure the REST Trigger Connection	717
Activate and Run the Recipe	717
Send Automatic Replies to Emails with a Specific Subject Using OCI AI Services	718
About This Recipe	718
System and Access Requirements	718
Before You Install the Recipe	718
Install and Configure the Recipe	719
Configure the Microsoft Outlook Connection	719
Configure the REST Trigger Connection	720
Configure the Integration Flows	720
Configure the Lookup Tables	721
Update Integration Properties	721
Activate and Run the Recipe	721
Send Compliance Documents from DocuSign to New ServiceNow Users	722
About This Recipe	722
System and Access Requirements	722
Before You Install the Recipe	723
Configure ServiceNow	723
Configure DocuSign	723
Install and Configure the Recipe	725
Configure the Oracle DocuSign Connection	725
Configure the Oracle ServiceNow Connection	726
Update the Integration Property	726
Activate and Run the Recipe	727
Send Files from OCI Object Storage to Oracle Autonomous Transaction Processing	728

About This Recipe	728
System and Access Requirements	729
Before You Install the Recipe	729
Configure Oracle Database Cloud Service	729
Configure OCI Object Storage	729
Configure OCI Notification Service	729
Install and Configure the Recipe	730
Configure the Oracle REST Connection	730
Configure the Oracle DBaaS Connection	731
Update the Integration Properties	731
Activate and Run the Recipe	732
Send Files from Your Application by Email	732
About This Recipe	732
System and Access Requirements	733
Install and Configure the Recipe	733
Configure the Oracle REST Trigger Connection	733
Configure the Lookup Table	733
Activate and Run the Recipe	733
Send New Appointment Details from an EMR Application to a Healthcare Provider	734
About This Recipe	734
System and Access Requirements	735
Before You Install the Recipe	735
Install and Configure the Recipe	736
Configure the MLLP Receiving Connection	736
Configure the Lookup Table	737
Activate and Run the Recipe	737
Send Notifications for Errors or Failures	738
About This Accelerator	738
System and Access Requirements	738
Install the Accelerator	738
Configure the Oracle Alerting Service Trigger Connection	739
Configure the Oracle Alerting Service Invoke Connection	739
Configure the Lookup Tables	739
Activate and Run the Accelerator	740
Send Notifications to Salesforce Lead Owners Using Twilio	741
About This Recipe	741
System and Access Requirements	741
Before You Install the Recipe	742
Configure Salesforce	742
Configure Twilio	745
Install and Configure the Recipe	746
Configure the Oracle Salesforce Connection	746

Configure the Oracle Twilio Connection	746
Update Integration Properties	747
Activate and Run the Recipe	747
Send Order Fulfillment Notifications from Oracle ERP Cloud Using Microsoft Office 365 Outlook	748
About This Recipe	748
System and Access Requirements	749
Before You Install the Recipe	749
Configure Oracle ERP Cloud	749
Configure Microsoft Email	749
Install and Configure the Recipe	750
Configure the Oracle ERP Cloud Connection	751
Configure the Oracle Microsoft Email Connection	751
Update Integration Properties	752
Activate and Run the Recipe	752
Send Salesforce Opportunity Documents to DocuSign	753
About This Recipe	753
System and Access Requirements	754
Before You Install the Recipe	754
Configure Salesforce	754
Configure DocuSign	758
Install and Configure the Recipe	759
Configure the Oracle Salesforce Connection	759
Configure the Oracle DocuSign Connection	760
Update Integration Properties	760
Activate and Run the Recipe	761
Send SurveyMonkey Surveys to Salesforce Contacts by Email	761
About This Recipe	762
System and Access Requirements	762
Before You Install the Recipe	762
Configure Salesforce	762
Configure SurveyMonkey	766
Install and Configure the Recipe	767
Configure the Salesforce Connection	767
Configure the SurveyMonkey Connection	768
Update Integration Properties	769
Activate and Run the Recipe	769
Send WhatsApp Notifications to Oracle ERP Cloud Lead Owners Using Twilio	770
About This Recipe	770
System and Access Requirements	771
Before You Install the Recipe	771
Configure Oracle ERP Cloud	771

Configure Twilio	772
Install and Configure the Recipe	772
Configure the Oracle ERP Cloud Connection	772
Configure the Twilio Connection	773
Set Up WhatsApp with Twilio	773
Update the Integration Property	773
Activate and Run the Recipe	774
Share YouTube Video Posts on Socials	774
About This Recipe	775
System and Access Requirements	775
Recipe Schema	775
Before You Install the Recipe	777
Create OCI Compute Instance	777
Install Oracle Integration Agent on Compute Instance	777
Create OCI Stream	777
Configure a Twitter Application	777
Create a LinkedIn Application	777
Create a Project in Google Cloud	777
Install and Configure the Recipe	778
Configure the YouTube REST Connection	778
Configure the Twitter Connection	778
Configure the LinkedIn Connection	779
Configure the Lookup Table	779
Activate and Run the Recipe	780
Sign and Verify Messages Using OCI Vault	780
About This Recipe	781
System and Access Requirements	781
Before You Install the Recipe	781
Install and Configure the Recipe	781
Configure the OCI Vault REST Connection	782
Configure the Lookup Table	782
Activate and Run the Recipe	783
Shortlist Resumes Automatically Using Slack and OpenAI	784
About This Recipe	784
System and Access Requirements	785
Before You Install the Recipe	785
Configure Slack	785
Install and Configure the Recipe	786
Configure the Slack Connection	787
Configure the REST Trigger Connection	787
Configure the Slack REST Connection	787
Configure the OpenAI Connection	788

Configure the Lookup Tables	788
Activate and Run the Recipe	789
Store and Manage Objects in Amazon S3	790
About This Recipe	790
Before You Install the Recipe	790
Configure Amazon S3	790
Install and Configure the Recipe	791
Configure the Amazon S3 Connection	791
Activate and Run the Recipe	792
Sync Accounts in Salesforce with Zendesk Organizations	792
About This Recipe	792
System and Access Requirements	793
Before You Install the Recipe	793
Configure Salesforce	793
Configure Zendesk	797
Install and Configure the Recipe	797
Configure the Salesforce Connection	797
Configure the Zendesk Connection	798
Update Integration Properties	798
Configure the Integration Flow	799
Activate and Run the Recipe	799
Sync Contacts Between Salesforce and Oracle B2C Service	800
About This Recipe	800
System and Access Requirements	800
Before You Install the Recipe	801
Configure Salesforce	801
Configure Oracle B2C Service	806
Install and Configure the Recipe	806
Configure the Salesforce Connection	806
Configure the Oracle B2C Service Connection	807
Activate and Run the Recipe	807
Sync Contacts Between SurveyMonkey and Oracle NetSuite	808
About This Recipe	808
System and Access Requirements	808
Before You Install the Recipe	808
Configure SurveyMonkey	808
Configure Oracle NetSuite	809
Install and Configure the Recipe	810
Configure the SurveyMonkey Connection	810
Configure the Oracle NetSuite Connection	810
Activate and Run the Recipe	811
Sync Contacts in Salesforce with Zendesk User Accounts	812

About This Recipe	812
System and Access Requirements	812
Before You Install the Recipe	813
Configure Salesforce	813
Configure Zendesk	816
Install and Configure the Recipe	816
Configure the Salesforce Connection	816
Configure the Zendesk Connection	816
Update Integration Property	817
Configure the Integration Flow	817
Activate and Run the Recipe	818
Sync Contracts between SAP Ariba and Oracle ERP Cloud	819
About This Recipe	819
System and Access Requirements	819
Before You Install the Recipe	819
Configure SAP Ariba	820
Configure Oracle ERP Cloud	820
Install and Configure the Recipe	821
Configure SAP Ariba	821
Configure the Oracle ERP Cloud Connection	821
Configure the Lookup Table	821
Activate and Run the Recipe	822
Sync Customer and Campaign Data Between Oracle Unity and Salesforce Marketing Cloud	823
About This Recipe	823
System and Access Requirements	823
Assumptions	824
Before You Install the Recipe	824
Configure Oracle Unity	824
Access Your FTP Server and Create a Directory	824
Configure Salesforce Marketing Cloud	824
Install and Configure the Recipe	827
Configure the Oracle Unity Connection	827
Configure the FTP Connection	828
Configure the REST Salesforce Marketing Cloud Connection	828
Configure the SOAP Salesforce Marketing Cloud Connection	829
Configure the REST OIC Invoke Connection	829
Configure the REST OIC Trigger Connection	830
Configure the Lookup Tables	830
Add or Edit the Attributes in Salesforce Marketing Cloud (Optional)	833
Add or Delete Attributes in Oracle Unity (Optional)	834
Add a New Email Campaign Event (Optional)	836
Activate and Run the Recipe	837

Sync Data Between Azure Storage and PostgreSQL Upon Receiving Notification on Azure Event Grid	839
About This Recipe	839
System and Access Requirements	840
Before You Install the Recipe	840
Access Your FTP Server	840
Configure Azure Event Grid	840
Configure Azure Storage (Get Blob)	841
Configure PostgreSQL	841
Install and Configure the Recipe	841
Configure the Azure Event Grid Connection	841
Configure the Azure Storage Invoke Connection	842
Configure the PostgreSQL Connection	842
Configure the FTP Connection	843
Activate and Run the Recipe	843
Sync Data Between Oracle CPQ Transactions and Salesforce Quotes	845
About This Recipe	845
System and Access Requirements	845
Before You Install the Recipe	846
Configure Oracle CPQ	846
Configure Salesforce	848
Install and Configure the Recipe	854
Configure the Oracle CPQ REST Connection	855
Configure the Oracle CPQ SOAP Connection	855
Configure the Salesforce Connection	856
Configure the Lookup Table	857
Configure the Integration Flow	857
Activate and Run the Recipe	857
Sync Email Accounts Between Okta and Oracle HCM Cloud	859
About This Recipe	859
System and Access Requirements	860
Before You Install the Recipe	860
Configure Oracle HCM Cloud	860
Configure Okta API Key	860
Install and Configure the Recipe	860
Configure the Oracle HCM Cloud Connection	860
Configure the REST Okta Connection	861
Activate and Run the Recipe	861
Sync Employee Identities Between Oracle HCM Cloud and an LDAP Instance	862
About This Recipe	862
System and Access Requirements	862
Before You Install the Recipe	862

Install and Configure the Recipe	863
Configure the Oracle HCM Cloud Connection	863
Configure the FTP Read Write Connection	863
Configure the Lookup Table	864
Activate and Run the Recipe	864
Sync ESG Data from EcoVadis to Suppliers in Oracle Procurement Cloud	865
About This Recipe	865
System and Access Requirements	865
Before You Install the Recipe	865
Configure Oracle Procurement Cloud	865
Configure EcoVadis	868
Install and Configure the Recipe	868
Configure the ERP Connection	868
Configure the ERP REST Connection	869
Configure the EcoVadis Connection	869
Configure the Lookup Table	870
Activate and Run the Recipe	871
Sync Incidents and Resources Between Oracle Field Service Cloud and Oracle B2C Service	871
About This Recipe	872
System and Access Requirements	872
Before You Install the Recipe	872
Configure Oracle B2C Service	872
Configure Oracle Field Service Cloud	873
Install and Configure the Recipe	873
Configure the Oracle B2C Service Connection	874
Configure the Oracle Field Service Cloud Connection	874
Activate and Run the Recipe	875
Sync Inventory Items Between Oracle Product Hub Cloud and Oracle E-Business Suite	876
About This Recipe	876
System and Access Requirements	876
Recipe Schema	876
Before You Install the Recipe	877
Configure Oracle ERP Cloud	877
Configure Oracle E-Business Suite	878
Install and Configure the Recipe	878
Configure the Oracle ERP Cloud Connection	878
Configure the Oracle E-Business Suite Connection	879
Configure the Lookup Table	879
Activate and Run the Recipe	880
Sync Jira Issues with GitHub	881
About This Recipe	881
System and Access Requirements	881

Before You Install the Recipe	881
Configure Jira	881
Configure GitHub	881
Install and Configure the Recipe	882
Configure the Jira Connection	882
Configure the GitHub Connection	882
Update Integration Properties	882
Activate and Run the Recipe	883
Sync Leads Between Marketo and Salesforce	884
About This Recipe	884
System and Access Requirements	884
Before You Install the Recipe	885
Configure Salesforce	885
Configure Marketo	889
Install and Configure the Recipe	889
Configure the Oracle Salesforce Connection	889
Configure the Oracle Marketo Connection	890
Configure the Lookup Table	890
Update Integration Properties	891
Activate and Run the Recipe	891
Sync New Cases from Microsoft Dynamics 365 to ServiceNow	892
About This Recipe	893
System and Access Requirements	893
Before You Install the Recipe	893
Configure Microsoft Dynamics 365	893
Configure ServiceNow	893
Install and Configure the Recipe	893
Configure the Microsoft Dynamics 365 Connection	894
Configure the ServiceNow Connection	894
Update Integration Properties	895
Activate and Run the Recipe	895
Sync Oracle NetSuite Customers with Oracle CX Sales and B2B Service Accounts	896
About This Recipe	896
System and Access Requirements	897
Before You Install the Recipe	897
Configure Oracle NetSuite	897
Configure Oracle CX Sales and B2B Service	898
Install and Configure the Recipe	900
Configure the Oracle NetSuite Connection	900
Configure the Oracle CX Sales and B2B Service Connection	901
Configure the Lookup Table	901
Activate and Run the Recipe	902

Sync Oracle NetSuite Items with QuickBooks Products	902
About This Recipe	903
System and Access Requirements	903
Before You Install the Recipe	903
Configure Oracle NetSuite	903
Configure QuickBooks	904
Install and Configure the Recipe	905
Configure the Oracle NetSuite Connection	905
Configure the Oracle QuickBooks Connection	906
Configure the Lookup Table	906
Activate and Run the Recipe	907
Sync Oracle Siebel Employees with Oracle Field Service Cloud Technicians	908
About This Recipe	908
System and Access Requirements	908
Before You Install the Recipe	908
Configure Oracle Siebel	908
Configure Oracle Field Service Cloud	909
Install and Configure the Recipe	909
Configure the Oracle SBL-OFS REST Connection	909
Configure the Oracle SBL-OFS OFSC Connection	909
Configure the Oracle SBL-OFS REST OFS Connection	910
Update Integration Properties	910
Activate and Run the Recipe	911
Sync Orders Between Oracle CX Sales and B2B Service and Oracle E-Business Suite	912
About This Recipe	912
System and Access Requirements	912
Before You Install the Recipe	912
Configure Oracle CX Sales and B2B Service	912
Configure Oracle E-Business Suite	913
Install and Configure the Recipe	914
Configure the Oracle CX Sales and B2B Service Connection	914
Configure the Oracle E-Business Suite Connection	915
Configure the Lookup Table	915
Activate and Run the Recipe	916
Sync Orders Between Shopify and SAP S/4HANA	917
About This Recipe	917
System and Access Requirements	917
Before You Install the Recipe	917
Configure Shopify	917
Configure OData for SAP S/4HANA	917
Install and Configure the Recipe	918
Configure the Shopify Connection	918

Configure the OData Connection for S/4HANA	919
Update Integration Properties	919
Configure the Lookup Tables	919
Activate and Run the Recipe	920
Sync Process Automation Audit Data to Oracle Autonomous Transaction Processing	921
About This Recipe	921
System and Access Requirements	922
Recipe Schema	922
Before You Install the Recipe	923
Configure Process Automation	923
Configure Oracle Autonomous Transaction Processing	924
Install and Configure the Recipe	924
Configure the Process Automation Connection	924
Configure the ATP Connection	925
Configure the REST Trigger Connection	926
Configure the Lookup Table	926
Activate and Run the Recipe	926
Sync Project Details from Oracle Primavera Cloud to PostgreSQL Using Amazon SQS	927
About This Recipe	927
System and Access Requirements	927
Before You Install the Recipe	927
Configure Oracle Primavera Cloud	928
Configure Amazon SQS	928
Configure PostgreSQL	928
Install and Configure the Recipe	929
Configure the Oracle Primavera Cloud Connection	929
Configure the Amazon SQS Connection	929
Configure the PostgreSQL Connection	930
Configure the Lookup Table	930
Activate and Run the Recipe	930
Sync Salesforce Accounts with Oracle CX Sales and Oracle Autonomous Transaction Processing	932
About This Recipe	932
System and Access Requirements	932
Before You Install the Recipe	933
Configure Salesforce	933
Configure Oracle Autonomous Transaction Processing	937
Install and Configure the Recipe	937
Configure the Salesforce Connection	937
Configure the Oracle ATP Connection	938
Configure the Oracle CX Sales Connection	939
Activate and Run the Recipe	939

Sync Salesforce Accounts with ServiceNow Companies	940
About This Recipe	940
System and Access Requirements	940
Before You Install the Recipe	940
Configure Salesforce	940
Configure ServiceNow	943
Install and Configure the Recipe	943
Configure the Oracle Salesforce Connection	943
Configure the Oracle ServiceNow Connection	944
Update Integration Property	944
Activate and Run the Recipe	944
Sync Salesforce Accounts with Shopify Customers	945
About This Recipe	945
System and Access Requirements	946
Before You Install the Recipe	946
Configure Salesforce	946
Configure Shopify	950
Install and Configure the Recipe	950
Configure the Oracle Salesforce Connection	950
Configure Shopify	951
Update Integration Property	951
Activate and Run the Recipe	951
Sync Salesforce Accounts with Workday Customers	953
About This Recipe	953
System and Access Requirements	953
Before You Install the Recipe	954
Configure Salesforce	954
Configure Workday	958
Configure Slack	958
Install and Configure the Recipe	958
Configure the Oracle Salesforce Connection	958
Configure the Oracle Workday Connection	959
Configure the Oracle Slack Connection	960
Configure the Lookup Tables	960
Update Integration Properties	961
Activate and Run the Recipe	961
Sync Salesforce Cases with ServiceNow Incidents	963
About This Recipe	963
System and Access Requirements	964
Before You Install the Recipe	964
Configure Salesforce	964
Configure ServiceNow	967

Install and Configure the Recipe	968
Configure the Oracle Salesforce Connection	968
Configure the Oracle ServiceNow Connection	969
Configure the Lookup Table	969
Activate and Run the Recipe	970
Sync Salesforce Contacts with ServiceNow Users	970
About This Recipe	971
System and Access Requirements	971
Before You Install the Recipe	971
Configure Salesforce	971
Configure ServiceNow	975
Install and Configure the Recipe	975
Configure the Oracle Salesforce Connection	975
Configure the Oracle ServiceNow Connection	976
Update Integration Property	976
Activate and Run the Recipe	977
Sync Stripe Customers with Salesforce Accounts	977
About This Recipe	977
System and Access Requirements	978
Before You Install the Recipe	978
Configure Stripe	978
Configure Salesforce	978
Install and Configure the Recipe	981
Configure the Stripe REST Trigger Connection	981
Configure the Oracle Salesforce Connection	981
Activate and Run the Recipe	982
Sync Supplier Updates Between Oracle Procurement Cloud and Oracle Autonomous Transaction Processing	982
About This Recipe	983
System and Access Requirements	983
Recipe Schema	983
Before You Install the Recipe	984
Configure Oracle ERP Cloud	984
Configure Apache Kafka	985
Configure Oracle Autonomous Transaction Processing	986
Install and Configure the Recipe	987
Configure the Oracle ERP Cloud Connection	987
Configure the Oracle Apache Kafka Connection	987
Configure the Oracle ATP Connection	988
Configure the Lookup Table	988
Activate and Run the Recipe	988
Sync Work Schedules between Workday and Oracle Field Service Cloud	989

About This Recipe	989
System and Access Requirements	990
Before You Install the Recipe	990
Access Your FTP Server	990
Configure Workday	990
Configure Oracle Field Service Cloud	992
Install and Configure the Recipe	992
Configure the Oracle FTP Connection	992
Configure the Oracle Workday Connection	993
Configure the Oracle Field Service Cloud Connection	993
Configure the Lookup Tables	994
Update the Report Owner ID	994
Activate and Run the Recipe	995
Sync Workday Employees with ServiceNow Users	997
About This Recipe	997
System and Access Requirements	998
Before You Install the Recipe	998
Configure Workday	998
Configure ServiceNow	1008
Install and Configure the Recipe	1008
Configure the Oracle REST Connection	1008
Configure the Oracle Workday Connection	1009
Configure the Oracle ServiceNow Connection	1009
Configure the Lookup Table	1010
Activate and Run the Recipe	1010
Sync Opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite Orders	1012
About This Recipe	1012
System and Access Requirements	1012
Recipe Schemas	1013
Before You Install the Recipe	1015
Configure Oracle CX Sales and B2B Service	1015
Configure Oracle NetSuite	1015
Install and Configure the Recipe	1016
Configure the Oracle CX Sales and B2B Service Connection	1016
Configure the Oracle NetSuite Connection	1017
Configure the Oracle REST Connection	1018
Activate and Run the Recipe	1018
Sync Oracle Service Cloud Incidents with Oracle CPQ Transactions	1019
About This Recipe	1019
System and Access Requirements	1020
Before You Install the Recipe	1020
Configure Oracle Service Cloud	1020

Configure Oracle CPQ	1021
Install and Configure the Recipe	1022
Configure the Oracle Service Cloud Connection	1022
Configure the Oracle CPQ Connection	1022
Update Integration Property	1023
Activate and Run the Recipe	1023
Validate ID for Loan Application Using OCI AI Services	1024
About This Recipe	1024
System and Access Requirements	1024
Before You Install the Recipe	1024
Configure Gmail	1024
Configure OCI Vision	1024
Configure OCI Generative AI	1025
Install and Configure the Recipe	1025
Configure the Gmail Connection	1025
Configure the REST Trigger Connection	1025
Configure the Integration Flow	1026
Configure the Lookup Tables	1026
Activate and Run the Recipe	1027
View Orders from Oracle ERP Cloud in Oracle CX Sales and B2B Service	1028
About This Recipe	1028
System and Access Requirements	1028
Before You Install the Recipe	1029
Configure Oracle ERP Cloud	1029
Configure Oracle CX Sales and B2B Service	1029
Install and Configure the Recipe	1030
Configure the Oracle ERP Cloud Connection	1030
Configure the Oracle CX Sales and B2B Service Connection	1030
Activate and Run the Recipe	1031

About This Content

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

Audience

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

Documentation Accessibility

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

Access to Oracle Support

Oracle 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.

Related Resources

For more information, see these Oracle resources:

- Oracle Integration documentation on the Oracle Help Center.
- Oracle Cloud at <http://cloud.oracle.com>.

Conventions

The following text conventions are used in this document.

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

1

Get Started

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

Topics:

- [Learn About Recipes and Accelerators](#)
- [Find Recipes and Accelerators](#)
- [Get More Details About Recipes and Accelerators](#)

Learn About Recipes and Accelerators

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

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

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

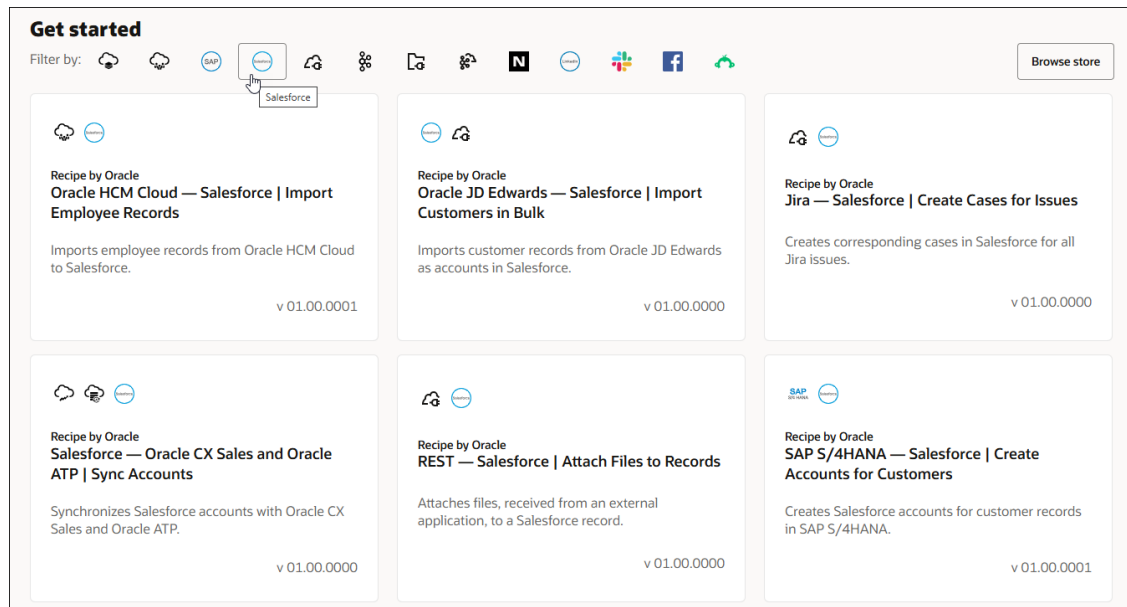
Here's a comparison of recipes and accelerators.

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

Find Recipes and Accelerators

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

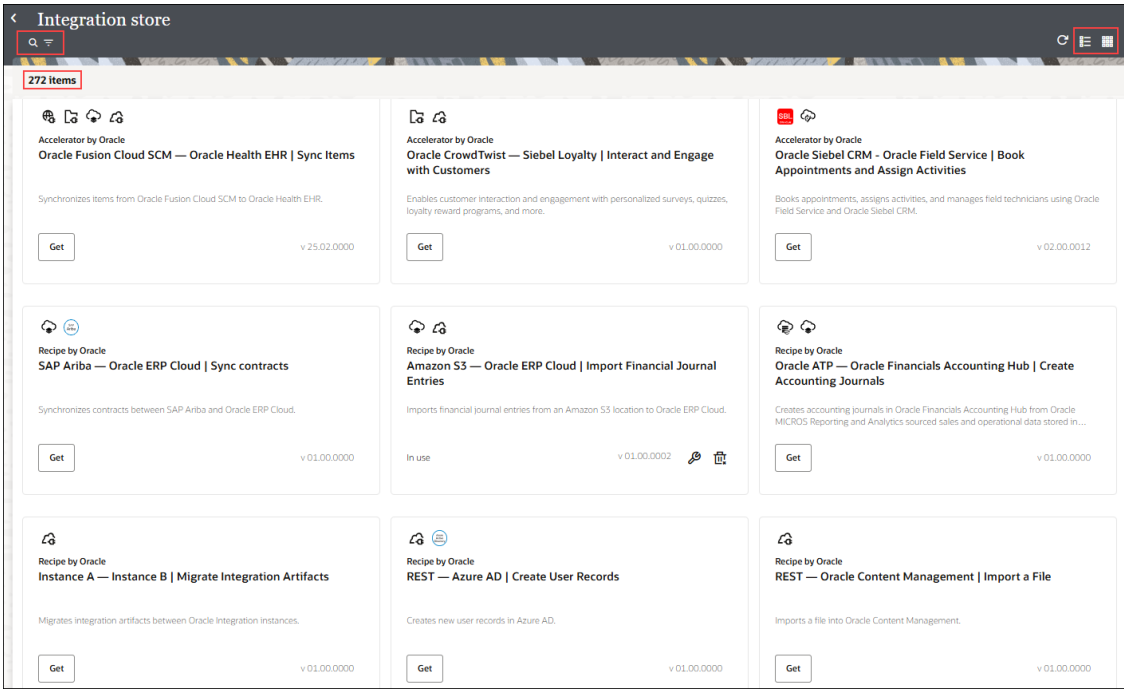
In the Get started section, you can filter the partial listing of recipes and accelerators according to applications used by clicking the product icon displayed on top. For example, to see some recipes and accelerators that use Salesforce, click the **Salesforce** icon.



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

Note

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



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

Search recipes and accelerators

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

Get More Details About Recipes and Accelerators

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

At a glance, you can see:

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

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

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

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

2

Install and Configure

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

Topics:

- [Install a Recipe or Accelerator](#)
- [Configure Resources](#)

Install a Recipe or Accelerator

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

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

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

After you've installed the recipe or accelerator, configure its resources. See [Configure Resources](#).

Uninstall a Recipe or Accelerator

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

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

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

To uninstall a recipe or accelerator:

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

Configure Resources

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

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

To configure resources of a recipe or accelerator using the wizard in the Configuration Editor,

click **Configure**  on the recipe or accelerator. See Edit Dependent Resources at the Project Level in *Using Integrations in Oracle Integration 3*.

Configure Resources in a Project:

1. In the Oracle Integration navigation pane, click **Projects**.

2. In the Projects page, click the project name or click **Edit** .

The project workspace opens displaying all the resources of the recipe or accelerator.

3. Within a section (for example, **Connections**), click **Actions** ... to view the options available for configuring the resource.

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

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

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

Add a resource

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

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

- Create an Integration
- Create a Connection.

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

- Create a Lookup

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

To configure resources for a package-based recipe or accelerator, see [Configure Resources in a Package](#).

Configure Resources in a Package

A few prebuilts in the Integration store are package-based. If your recipe or accelerator is a package-based prebuilt, then you configure its resources in the Configuration Editor.

1. Click **Configure**  on a recipe or accelerator card of a package-based prebuilt.
The Configuration Editor appears.
2. View the details on the editor.
The title bar includes the name of the recipe or accelerator. Click **Connections**, **Lookups**, **Activation**, **Certificates**, and **Libraries** to view the resources that are included in the package.
3. Hover over a row to reveal the options for configuring, editing, or replacing a resource.

Option on the menu	Available for	Description
Add schedule	Integrations	Define when the integration flow runs. This option is available only if the integration flow uses a Scheduled Orchestration style. Click Actions  on an integration flow to reveal this option.
Update property values	Integrations	Update integration property values for any integration flow in which properties have been defined. Click Actions  on an integration flow to reveal this option.
Edit	Integrations Connections Certificates Lookups Libraries	Access the corresponding page for editing the integration flow or any other resource. For more information about editing these resources, see: • Configure Connection Properties. • Manage Security Certificates. • Map Data and Create Lookups. • Use Libraries to Manage Functions for Integrations.

Option on the menu	Available for	Description
Replace	Connections Certificates	Replace a connection or a certificate (PGP key) across all the integration flows 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 Revert is available.
Open Details	Connections Certificates Lookups Libraries	Display the number and name of the integration flows in this package that use the selected resource.

4. Make the necessary edits to the resources and save your changes, then click **Go back** to return to the Configuration Editor.
5. Continue to edit each resource. When you are ready, you can activate the package from the Configuration Editor or the Integration Store.

3

Activate and Monitor

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

- [Activate a Recipe or Accelerator](#)
- [Monitor Integrations](#)

Activate a Recipe or Accelerator

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

To activate a recipe or accelerator:

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

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

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

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

Monitor Integrations

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

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

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

4

Explore Recipes and Accelerators

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

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

Note

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

Browse All

Browse all recipes and accelerators in this guide:

- [Approve Invoices Using OCI AI Services](#)
- [Attach Files to Salesforce Records](#)
- [Automate Customer Feedback Processing Using OCI AI Services and ServiceNow](#)
- [Calculate Total Balance](#)
- [Concatenate Messages](#)
- [Copy List of Files on FTP Server](#)
- [Copy Objects from OCI Object Storage to Microsoft SharePoint](#)
- [Create Accounting Journals in Oracle Accounting Hub](#)
- [Create Accounts in Oracle CX Sales and B2B Service for HubSpot Companies](#)
- [Create Activities in Oracle Field Service Cloud for Oracle ERP Cloud Work Orders](#)
- [Create Activities in Oracle Primavera Cloud for Jira Issues](#)
- [Create Advance Shipment Notices in Oracle Fusion Cloud SCM](#)
- [Create Alerts in Microsoft Teams for Jira Issues](#)
- [Create and Add a User to Multiple Groups in OCI](#)
- [Create Box Folders for ServiceNow Incidents](#)
- [Create Box Folders for Zendesk Organizations](#)
- [Create Customer Records in Oracle ERP Cloud](#)
- [Create Customer Records in Oracle NetSuite](#)
- [Create Customers in Magento Commerce Cloud for Oracle ERP Cloud Accounts](#)

- [Create Customers in Shopify for Salesforce Accounts](#)
- [Create Invoices in Oracle ERP Cloud Using OCI Document Understanding AI Service](#)
- [Create Jira Issues for Exceptions in Integrations](#)
- [Create Jira Issues for Project Tasks in Oracle NetSuite](#)
- [Create Jira Issues for ServiceNow Incidents](#)
- [Create New User Records in Azure Active Directory](#)
- [Create Oracle CPQ Transactions for Oracle Intelligent Advisor Interviews](#)
- [Create Orders in Oracle Fusion Cloud Service for Oracle CPQ Opportunities](#)
- [Create Products in Magento Commerce Cloud and QuickBooks for Oracle ERP Cloud Items](#)
- [Create Products in Shopify for Items in Oracle ERP Cloud](#)
- [Create Project Tasks in Oracle NetSuite for Jira Issues](#)
- [Create Sales Orders in Oracle ERP Cloud for Salesforce Opportunities](#)
- [Create Sales Orders in Oracle NetSuite for Invoices in Magento Commerce Cloud](#)
- [Create Sales Orders in Oracle NetSuite for Salesforce Opportunities](#)
- [Create Sales Receipts in QuickBooks for Oracle NetSuite Invoices](#)
- [Create Salesforce Accounts for SAP ERP Customers](#)
- [Create Salesforce Accounts for SAP S/4HANA Customers](#)
- [Create Salesforce Cases for Jira Issues](#)
- [Create SAP ERP Orders for Closed Salesforce Opportunities](#)
- [Create Suppliers in SAP Ariba for Vendors in SAP ERP](#)
- [Create Users in OCI Identity and Access Management for Persons in PeopleSoft](#)
- [Create Workers in Oracle HCM Cloud](#)
- [Create Zendesk Tickets for Jira Issues](#)
- [Create Zendesk Tickets for Salesforce Cases](#)
- [Delete Old Files](#)
- [Encrypt and Decrypt Data Using OCI Vault](#)
- [Export Campaign Data in Bulk from Snowflake to Salesforce Marketing Cloud Using GCP Storage](#)
- [Export Data in Bulk from Oracle ERP Cloud to an FTP Server](#)
- [Export Employee Images from Oracle HCM Cloud to an FTP Server](#)
- [Export Employee Records from Workday to Azure Active Directory](#)
- [Export Employee Records from Workday to SAP](#)
- [Export Google Gmail Attachments to an FTP Server](#)
- [Export Invoices from PayPal to an FTP Server](#)
- [Export Items in Bulk from Oracle ERP Cloud to Shopify](#)
- [Export Items with Variants from Oracle ERP Cloud to Shopify](#)
- [Export Messages from Apache Kafka to Amazon S3](#)

- [Export New Employee Data from Oracle HCM Cloud to an FTP Server](#)
- [Export New Hire Information from Oracle HCM Cloud](#)
- [Export Oracle NetSuite Records to QuickBooks](#)
- [Export Purchase Orders from Oracle ERP Cloud to FTP Server](#)
- [Export Workday Employees to an FTP Server](#)
- [Extract Approved Expense Reports from SAP Concur and Write to an FTP Server](#)
- [Extract Data from Unstructured Files Using OCI Vision](#)
- [Extract Document Information using OCI AI Services](#)
- [Fetch Data in Chunks from Oracle Autonomous Transaction Processing](#)
- [Fusion HCM License and Certification Validation by EverCheck](#)
- [Fusion HCM License and Certification Validation by HireRight](#)
- [Generate an ANSI 834 Benefits File from Oracle HCM Cloud Extract File](#)
- [Generate and Email Patient Report Summary Using OCI AI Services](#)
- [Generate Oracle Hospitality Access Tokens](#)
- [Get Details of a Facebook Page](#)
- [Get Incident Details From Oracle B2C Service](#)
- [Get Opportunity Details from Oracle B2B Service](#)
- [Get Organization Details from Oracle Service Cloud](#)
- [Get Stocks Data](#)
- [Get Weather Data](#)
- [Handle Throttling Using Parking Lot Pattern](#)
- [Import a File into Oracle Content Management](#)
- [Import a Payment Request into Oracle ERP Cloud](#)
- [Import Absence Records from Oracle HCM Cloud to Google Calendar](#)
- [Import Absence Records from Oracle HCM Cloud to Microsoft Calendar](#)
- [Import and Update Files in Bulk](#)
- [Import Candidate Records from an FTP Server to Taleo](#)
- [Import Contacts from Oracle Autonomous Transaction Processing to Oracle Responsys](#)
- [Import Customer Records from an FTP Server to Oracle ERP Cloud](#)
- [Import Customer Records from Shopify to Stripe](#)
- [Import Customer Records in Bulk from Oracle JD Edwards to Salesforce](#)
- [Import Customers and Credit Card Transactions from SAP ASE \(Sybase\) to Oracle Unity](#)
- [Import Documents from Oracle ERP Cloud to Adobe Sign](#)
- [Import Documents from Salesforce to Adobe Sign](#)
- [Import Employee Records from an FTP Server to Oracle HCM Cloud](#)
- [Import Employee Records from Oracle HCM Cloud to Jira](#)
- [Import Employee Records from Oracle HCM Cloud to Oracle E-Business Suite](#)
- [Import Employee Records from Oracle HCM Cloud to Salesforce](#)

- [Import Employee Records from Oracle HCM Cloud to Zendesk](#)
- [Import Employee Vacation and Delegation Rules Report from Oracle HCM Cloud to Oracle Autonomous Transaction Processing](#)
- [Import Employees from SAP SuccessFactors to Oracle E-Business Suite](#)
- [Import Files from an FTP Server to Oracle EPM Cloud](#)
- [Import Files from an FTP Server to Oracle NetSuite](#)
- [Import Files from File Server to OCI Object Storage](#)
- [Import Financial Journal Entries from Amazon S3 to Oracle ERP Cloud](#)
- [Import Financial Journal Entries from an FTP server to Oracle ERP Cloud](#)
- [Import Leads from SugarCRM to Oracle ERP Cloud](#)
- [Import Marketo Leads to Oracle Autonomous Data Warehouse](#)
- [Import Offer Letters from Oracle HCM Cloud to Adobe Sign](#)
- [Import Oracle Commerce Cloud Orders to Oracle Intelligent Advisor Interviews](#)
- [Import Orders from Oracle ERP Cloud to Oracle Autonomous Data Warehouse](#)
- [Import Orders from Shopify to Oracle ERP Cloud](#)
- [Import Orders from Shopify to Salesforce](#)
- [Import Orders from Zuora to a Hasura Supported Database](#)
- [Import Products from Salesforce to Shopify](#)
- [Import Purchase Orders from Oracle ERP Cloud to Microsoft SQL Server](#)
- [Import Purchase Orders from Oracle JD Edwards EnterpriseOne to Oracle ERP Cloud](#)
- [Import Sales Orders from Oracle ERP Cloud to MySQL](#)
- [Import Sales Orders from Oracle NetSuite to Oracle Autonomous Data Warehouse](#)
- [Import Segments from Oracle Unity to Salesforce Marketing Cloud](#)
- [Import Supplier Records Asynchronously from Oracle ERP Cloud to Oracle Autonomous Transaction Processing](#)
- [Import Supplier Records from SAP S/4HANA to Oracle ERP Cloud](#)
- [Import Suppliers from File Server to Oracle ERP Cloud](#)
- [Import Users from Oracle HCM Cloud to Okta](#)
- [Import Worker Data from an FTP Server to Oracle HCM Cloud](#)
- [Increment Variable in a Loop](#)
- [Integrate DiCentral with Oracle SCM Cloud](#)
- [Integrate KanbanBOX with Oracle SCM Cloud](#)
- [Integrate Oracle Fusion Cloud IoT Intelligent Applications with Oracle Field Service Cloud](#)
- [Integrate Oracle MICROS Reporting and Analytics Sales Data from ATP with Oracle Financials Accounting Hub](#)
- [Integrate Oracle MICROS Reporting and Analytics with Oracle Autonomous Transaction Processing](#)
- [Integrate with DocuSign](#)
- [Interact with LinkedIn](#)

- [Interact with Slack](#)
- [Interact with Twitter](#)
- [Invoke a Local Integration](#)
- [Invoke a Non-Local Integration](#)
- [Invoke Pop-Up Notifications for Events in Oracle Fusion Applications](#)
- [Log a Request Message](#)
- [Manage Objects in OCI Object Storage](#)
- [Manage ServiceNow Incidents](#)
- [Migrate Integration Artifacts Between Instances](#)
- [Perform Arithmetic Operations](#)
- [Perform Multiple Operations on Resources](#)
- [Post Slack Notifications for Completed SurveyMonkey Surveys](#)
- [Post Slack Notifications for New Marketo Leads](#)
- [Post Slack Notifications for Oracle CPQ Quote Approvals](#)
- [Post Slack Notifications for Oracle ERP Cloud Purchase Order Approvals](#)
- [Post Slack Notifications for ServiceNow Activities](#)
- [Post Slack Notifications for Stripe Charges](#)
- [Process Inventory, Order, and Shipping Info Between Oracle ERP/SCM Cloud and Oracle WMS Cloud](#)
- [Publish and Subscribe to Oracle Cloud Infrastructure Streaming Service](#)
- [Publish Shopify Orders to GCP Pub Sub](#)
- [Read and Write Files on an FTP Server](#)
- [Resequence and Process Messages in a Specific Order](#)
- [Send a Hello Message](#)
- [Send Automatic Replies to Emails with a Specific Subject Using OCI AI Services](#)
- [Send Compliance Documents from DocuSign to New ServiceNow Users](#)
- [Send Files from OCI Object Storage to Oracle Autonomous Transaction Processing](#)
- [Send Files from Your Application by Email](#)
- [Send New Appointment Details from an EMR Application to a Healthcare Provider](#)
- [Send Notifications for Errors or Failures](#)
- [Send Notifications to Salesforce Lead Owners Using Twilio](#)
- [Send Order Fulfillment Notifications from Oracle ERP Cloud Using Microsoft Office 365 Outlook](#)
- [Send Salesforce Opportunity Documents to DocuSign](#)
- [Send SurveyMonkey Surveys to Salesforce Contacts by Email](#)
- [Send WhatsApp Notifications to Oracle ERP Cloud Lead Owners Using Twilio](#)
- [Share YouTube Video Posts on Socials](#)
- [Sign and Verify Messages Using OCI Vault](#)

- [Shortlist Resumes Automatically Using Slack and OpenAI](#)
- [Store and Manage Objects in Amazon S3](#)
- [Sync Accounts in Salesforce with Zendesk Organizations](#)
- [Sync Contacts Between Salesforce and Oracle B2C Service](#)
- [Sync Contacts Between SurveyMonkey and Oracle NetSuite](#)
- [Sync Contacts in Salesforce with Zendesk User Accounts](#)
- [Sync Contracts between SAP Ariba and Oracle ERP Cloud](#)
- [Sync Customer and Campaign Data Between Oracle Unity and Salesforce Marketing Cloud](#)
- [Sync Data Between Azure Storage and PostgreSQL Upon Receiving Notification on Azure Event Grid](#)
- [Sync Data Between Oracle CPQ Transactions and Salesforce Quotes](#)
- [Sync Email Accounts Between Okta and Oracle HCM Cloud](#)
- [Sync Employee Identities Between Oracle HCM Cloud and an LDAP Instance](#)
- [Sync ESG Data from EcoVadis to Suppliers in Oracle Procurement Cloud](#)
- [Sync Incidents and Resources Between Oracle Field Service Cloud and Oracle B2C Service](#)
- [Sync Inventory Items Between Oracle Product Hub Cloud and Oracle E-Business Suite](#)
- [Sync Jira Issues with GitHub](#)
- [Sync Leads Between Marketo and Salesforce](#)
- [Sync New Cases from Microsoft Dynamics 365 to ServiceNow](#)
- [Sync Oracle NetSuite Customers with Oracle CX Sales and B2B Service Accounts](#)
- [Sync Oracle NetSuite Items with QuickBooks Products](#)
- [Sync Oracle Siebel Employees with Oracle Field Service Cloud Technicians](#)
- [Sync Orders Between Oracle CX Sales and B2B Service and Oracle E-Business Suite](#)
- [Sync Orders Between Shopify and SAP S/4HANA](#)
- [Sync Process Automation Audit Data to Oracle Autonomous Transaction Processing](#)
- [Sync Project Details from Oracle Primavera Cloud to PostgreSQL Using Amazon SQS](#)
- [Sync Salesforce Accounts with Oracle CX Sales and Oracle Autonomous Transaction Processing](#)
- [Sync Salesforce Accounts with ServiceNow Companies](#)
- [Sync Salesforce Accounts with Shopify Customers](#)
- [Sync Salesforce Accounts with Workday Customers](#)
- [Sync Salesforce Cases with ServiceNow Incidents](#)
- [Sync Salesforce Contacts with ServiceNow Users](#)
- [Sync Stripe Customers with Salesforce Accounts](#)
- [Sync Supplier Updates Between Oracle Procurement Cloud and Oracle Autonomous Transaction Processing](#)
- [Sync Work Schedules between Workday and Oracle Field Service Cloud](#)

- [Sync Workday Employees with ServiceNow Users](#)
- [Sync Opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite Orders](#)
- [Sync Oracle Service Cloud Incidents with Oracle CPQ Transactions](#)
- [Validate ID for Loan Application Using OCI AI Services](#)
- [View Orders from Oracle ERP Cloud in Oracle CX Sales and B2B Service](#)

Approve Invoices Using OCI AI Services

Use this recipe to review invoice approval, notify manager, and create invoices in Oracle ERP Cloud using the artificial intelligence capabilities of the following OCI AI services - Oracle Cloud Infrastructure Document Understanding (OCI Document Understanding), Oracle Cloud Infrastructure Generative AI (OCI Generative AI) and Oracle Cloud Infrastructure Generative AI Agents (OCI Generative AI Agents).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe receives invoices as PDF files and uses OCI Document Understanding, OCI Generative AI, and OCI Generative AI Agents to review invoice approval, notify managers, and create invoices in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When the main integration flow receives an invoice PDF file, it encodes the PDF file to base64, and sends the base64 encoded file to OCI Document Understanding. Using the artificial intelligence capabilities of OCI Document Understanding, information is extracted from the base64 encoded invoice file. The result of OCI Document Understanding invokes the child integration flow. An invoice report is generated leveraging the artificial intelligence capabilities of OCI Generative AI Agents retrieval augmented generation (RAG). Subsequently, the invoice report is sent to OCI Generative AI in order to get the approval status of the invoice.

Based on the approval status of the invoice, the recipe performs one of the following:

- Creates a corresponding invoice in Oracle ERP Cloud, if the invoice complies with company policy and is approved.
- Notifies manager if further approval is required for the invoice.
- Sends notification if the invoice is rejected.

Note

To successfully run this recipe, the input PDF file have to be in a certain format. You can download the following file for reference: [Sample file](#).

System and Access Requirements

- Oracle Integration 3
- OCI Document Understanding
- OCI Generative AI
- OCI Generative AI Agents
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your OCI Document Understanding, OCI Generative AI and OCI GenAI Agents and Oracle ERP Cloud instances in order to successfully review invoice approval, notify managers and create corresponding invoices in Oracle ERP Cloud.

Configure OCI Document Understanding

You must configure and apply policies at the compartment level for OCI Document Understanding users.

See **Apply a policy to grant `MANAGE` permission** under [Policy to Grant Users Access to Document Understanding APIs](#) in *OCI Document Understanding* documentation.

Ensure that you select the same OCI compartment (where you applied the Document Understanding policies) when you [configure the integration flow](#) of the recipe to use the **OCI Document Understanding** action.

Configure OCI Generative AI

You must configure and apply policies at the compartment level for OCI Generative AI users.

See [Getting Access to Generative AI](#) in *OCI Generative AI* documentation.

Ensure that you select the same OCI compartment (where you applied the OCI Generative AI policies) when you [configure the integration flow](#) of the recipe to use the **OCI Generative AI** action.

Configure OCI Generative AI Agents

You must configure and apply policies at the compartment level for OCI Generative AI Agents users.

See [Getting Access to Generative AI Agents](#) in *OCI Generative AI Agents* documentation.

Ensure that you select the same OCI compartment (where you applied the OCI Generative AI Agents policies) when you [configure the integration flow](#) of the recipe to use the **OCI Generative AI Agents RAG** action.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign required roles to the user account. For more information on the roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ERP Cloud Connection

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

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

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

A message confirms if your test is successful.

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

Configure the REST Trigger Connection

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

A message confirms if your test is successful.

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

Configure the Integration Flows

Edit the **Validate Invoice Against Corporate Policy** and **Retrieve Data from GenAI RAG** integration flows to configure actions for the following OCI AI services.

- [OCI Document Understanding](#)
- [OCI Generative AI](#)
- [OCI Generative AI Agents](#)

OCI Document Understanding

1. In the Integrations section, click the **Validate Invoice Against Corporate Policy** integration flow.
2. On the integration canvas, click the **OCI Document Understanding** element, and then click **Actions**  and select **Edit**.
3. Click **Next** and select the OCI compartment.
4. Click **Next** and then **Done**.
5. Click **Save** to save the integration.
6. To return to the project workspace, click **Go back** .

To know more about the document understanding action, see Extract Document Information with a Document Understanding Action in *Using Integrations in Oracle Integration 3*.

OCI Generative AI

1. In the Integrations section, click the **Validate Invoice Against Corporate Policy** integration flow.
2. On the integration canvas, click the **OCI Generative AI** element, and then click **Actions**  and select **Edit**.
3. Click **Next** and select the OCI region, OCI compartment, model, and model ID.
4. Click **Next** and then **Done**.
5. Click **Save** to save the integration.
6. To return to the project workspace, click **Go back** .

To know more about the generative AI action, see Generate Text with a Generative AI Action in *Using Integrations in Oracle Integration 3*.

OCI Generative AI Agents

1. In the Integrations section, click the **Retrieve Data from GenAI RAG** integration flow.
2. On the integration canvas, click the **OCI Generative AI Agents RAG** element for List agent endpoints action (ListRAGAgentEndpoints), and then click **Actions**  and select **Edit**.
3. Click **Next** and select the OCI region and OCI compartment.

4. Click **Next** and then **Done**.
5. On the integration canvas, click the **OCI Generative AI Agents RAG** element for Create session action (RAGbasedChatSession), and then click **Actions** ... and select **Edit**.
6. Click **Next** and select the OCI region.
7. Click **Next** and then **Done**.
8. Click **Save** to save the integration.
9. To return to the project workspace, click **Go back** .

To know more about the generative AI agents RAG action, see Ask Questions and Receive Answers Based on Current Enterprise Data with a Generative AI Agents RAG Action in *Using Integrations in Oracle Integration 3*.

Configure the Lookup Table

Edit the lookup key in the **ManagerEmail_Lookup** lookup table to enter the email address of the manager to whom notifications have to be sent for invoice approval.

1. In the Lookups section, click the lookup name.
2. In the **value** column, enter the email address of the manager.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the **Validate Invoice Against Corporate Policy** integration flow, browse and upload the invoice PDF file, and then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flows in Oracle Integration. See [Monitor Integrations](#).
4. If the invoice is approved, log in to your Oracle ERP Cloud and check if a corresponding new invoice is created.
5. If further approval is required for the invoice, check if the manager received an approval notification email.
6. If the invoice is rejected, check if a notification regarding the same has been sent.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*
- [OCI Document Understanding](#)
- [OCI Generative AI](#)
- [OCI Generative AI Agents](#)

Attach Files to Salesforce Records

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

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

System and Access Requirements

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

Before You Install the Recipe

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

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

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

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

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
 - a. On the Salesforce Setup page, expand **Manage Users** under the Administer section in the left navigation pane.
 - b. Click **Profiles**.
 - c. On the Profiles page, click **New Profile**.
 - d. On the resulting page:
 - i. Select **Standard User** in the **Existing Profile** field.
 - ii. Enter a name for the new profile, for example, **API Enabled**, and click **Save**.
The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - e. Click **Edit** on the Profile Detail page.
 - f. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

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

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:

- i. Enter a first name and last name for the user, for example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field.
Note down this user name.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce.
- You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note down the same.
- Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
- i. Stay signed in as the new user, and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account.
Note down the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version.
- a. On the Salesforce Setup page, scroll to the Build section in the left navigation pane.
 - b. Click **Develop**, and then **API**.
 - c. On the API WSDL page, click the **Generate Enterprise WSDL** link.
- The WSDL is displayed in a new browser tab, and your current API version is present in the second line. For example:

Salesforce.com Enterprise Web Services API Version 52.0

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

Note

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

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

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

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle REST Trigger Connection

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

A message confirms if your test is successful.

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

Configure the Oracle Salesforce Connection

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

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.

Field	Information to Enter
API Version	Enter your current Salesforce API version. To obtain the API version, see Before You Install the Recipe .

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

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the user name of the account created for Oracle Integration on Salesforce. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on Salesforce.

Note

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

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#)
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the necessary file and the ID of the Salesforce record to which you want to attach the file. Upload the file in the POST request's **Body** field and specify the Salesforce record's ID as a template parameter at the end of the integration's endpoint URL, replacing the placeholder {salesforceObjectId}.

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

3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

- b. On the Configure and run page, in the Request section, enter the following details:
 - i. On the **URI Parameters** tab, against the **salesforceObjectId** field, enter the ID of the Salesforce record to which you want to attach the file. To obtain the Salesforce ID of a record, see [Before You Install the Recipe](#).
 - ii. On the **Body** tab, upload a file from your computer.
 - c. Click **Run**.

The integration flow now attaches the file you uploaded to the specified Salesforce record. The Activity Stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the test page, you should see the status as **200 OK** and **"success" : true**.
4. Log in to your Salesforce instance as an Administrator and check if the file has been attached to the specified record. Because we've used an account record as an example in this document, the following steps detail how to navigate to an account record on Salesforce and check for the file attachment.
 - a. On the Salesforce Setup page, click the **Accounts** tab.

Note

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

- b. On the Accounts Home page, click the account whose ID you specified in the REST request.
- c. On the account's page, scroll to the Notes & Attachments section.

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

Related Documentation

- [Using the Salesforce Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)

Automate Customer Feedback Processing Using OCI AI Services and ServiceNow

Use this recipe to automate the process of analyzing customer email feedback related to products and support using OCI AI services and ServiceNow.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe automates the process of analyzing customer email feedback related to products and support. Using OCI Language and OCI Generative AI, the recipe evaluates sentiment and key aspects of incoming emails and takes appropriate actions based on the content.

The automation helps businesses to understand customer sentiment in real time using OCI Language, automatically route feedback to the relevant departments (Marketing, Support, and so on), generate professional email responses using OCI Generative AI, and create ServiceNow incidents for negative feedback.

System and Access Requirements

- An account on Microsoft Azure with the Administrator role
- Oracle Integration 3
- An account on ServiceNow with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks in Microsoft Office 365 Outlook, OCI Generative AI, OCI Language, and ServiceNow.

1. To configure Microsoft Office 365 Outlook, see Prerequisites for Creating a Connection in *Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*.
2. To configure OCI Generative AI and OCI Language, see Prerequisites in *Using Integrations in Oracle Integration 3*.
3. To configure ServiceNow, see Prerequisites for Creating a Connection in *Using the ServiceNow Adapter with Oracle Integration 3*.

Install and Configure the Recipe

Install the recipe on your Oracle Integration instance.

See [Install a Recipe or Accelerator](#).

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

Configure the Microsoft Office 365 Outlook Connection

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

Field	Information to Enter
Client Id	Enter the client Id.
Client Secret	Enter the client secret.
Scope	Enter the scope URL. For example: • https://graph.microsoft.com/Mail.ReadWrite • https://graph.microsoft.com/Mail.Send offline_access

3. Click **Provide Consent**.

This enables Oracle Integration to interact with the Microsoft Office 365 Outlook account used to create the application at <https://portal.azure.com/>.

4. If everything is correct, you are prompted for the Oracle Integration credentials. These are the same credentials you use to log in to Oracle Integration.
 - a. Enter the credentials and click **OK**.
 - b. Enter the Microsoft Office 365 Outlook account credentials.

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

The Access Allowed! page is displayed.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

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

Configure the ServiceNow Connection

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

Field	Information to Enter
Security policy	Select Basic Authentication from the drop-down list.
Username	Enter your ServiceNow username.
Password	Enter your ServiceNow password.

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

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

Configure the Integration Flow

Edit the integration flows to configure actions for the following OCI AI services.

- [OCI Language](#)
- [OCI Generative AI](#)

OCI Language

1. In the Integrations section, click the **Read Support Emails** integration flow.
2. On the integration canvas, click the **OCI Language** element, and then click **Actions . . .** and select **Edit**.

3. Click **Action** and select **Sentiment analysis**, and then click **Continue**.
4. From the **Compartment** drop-down list, select the OCI compartment.
5. Click **Next** and then **Done**.
6. Click **Save** to save the integration.
7. To return to the project workspace, click **Go back** .

To know more about the language action, see Perform Text Analysis and Translation with a Language Action in *Using Integrations in Oracle Integration 3*.

OCI Generative AI

1. In the Integrations section, click the **Send Email with GenAI** integration flow.
2. On the integration canvas, click the **OCI Generative AI** element, and then click **Actions**  and select **Edit**.
3. From the **Select Categories** drop-down list, select **On demand**.
4. Click **Action** and select **Chat**, and then click **Continue**.
5. Select the OCI region, OCI compartment, model, and model ID.
6. Click **Continue** and then **Finish**.
7. Click **Save** to save the integration.
8. To return to the project workspace, click **Go back** .

To know more about the generative AI action, see Generate Text with a Generative AI Action in *Using Integrations in Oracle Integration 3*.

Configure the Lookup Table

The recipe contains the following two lookup tables that you can edit as necessary. Edit the lookup tables to enter appropriate values for the lookup keys.

- **SENTIMENT_RESPONSE_LOOKUP**: Use this lookup for mapping sentiment aspects to corresponding responses. For example, a aspect sentiment of Support_Positive can be mapped to a response of SupportEmail.

Aspect_Sentiment	Response: Action
Support_Positive	SupportEmail: Send an email to the support email address.
Support_Negative	ServiceCloud: Create an incident in ServiceNow.
Product_Positive	MarketingEmail: Send an email to the marketing email address.
Product_Negative	ServiceCloud: Create an incident in ServiceNow.
resolution_Positive	MarketingEmail: Send an email to the marketing email address.
resolution_Negative	SupportEmail: Send an email to the support email address.
product quality_Negative	ServiceCloud: Create an incident in ServiceNow.
experience_Negative	SupportEmail: Send an email to the support email address.

- **EmailConfiguration**: Use this lookup for configuring the support and marketing emails.

Key	Value
SupportEmail	Specify the support email address.
MarketingEmail	Specify the marketing email address.

1. In the Lookups section, click the lookup name.
2. In the **Property_Value** column, enter an email ID of your choice.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Read Support Emails** integration flow, then select **Run**.

The Configure and run page is displayed with the following parameter. Update the parameter with appropriate value.

- maildate: The date from which emails should be read.

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. For positive feedback, verify if professional email responses are generated using OCI Generative AI. For any negative feedback, verify if the incident is created in ServiceNow.

Related Documentation

- [Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3](#)
- [Using the ServiceNow Adapter with Oracle Integration 3](#)

Calculate Total Balance

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)

- [Activate and Run the Recipe](#)

About This Recipe

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

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

System and Access Requirements

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

Before You Install the Recipe

To create a connection to the SOAP service, perform the configuration steps described in [Create a SOAP Adapter Connection](#).

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle SOAP Trigger Connection

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#)
2. Run the recipe from an external application.

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

For example:

CARD NUMBER: 5105 1051 0510 5100

BALANCE: 151

LOAN AMOUNT: 100

CREDIT RATE: 4

- e. Authenticate your request.

The integration flow of the recipe is triggered.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#)
4. Verify the result which should be the total balance of the credit cards.

Related Documentation

- [Using the SOAP Adapter with Oracle Integration 3](#)

Concatenate Messages

Use this recipe to concatenate messages using the JavaScript callout function.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to invoke and use a JavaScript callout activity to concatenate messages.

To use the recipe, you must install the recipe and configure the connection and other resources within it. The recipe uses the REST adapter in an orchestrated integration. The integration flow is triggered when you specify a URL with two string query parameters. These parameters are then concatenated by a JavaScript callout function and a string message is returned as a response.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Rest Sample Trigger Connection

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a GET request to this endpoint URL along with the request message data.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the request messages in the query parameter fields for message1 and message2.
 - c. Click **Run**.

The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success status, 200 OK, and the response returned.
4. View the request and response messages in the Activity stream pane.

- Click **View payload**  on the **Logger logMessage** item to view the request message.
- Click **View payload**  on the last item of the activity stream to view the response message.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Copy List of Files on FTP Server

Use this recipe to copy a list of files from one directory to another on an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe uses the FTP Adapter to copy a list of files from one directory to another on an FTP server, and modify their content in Oracle Integration.

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 3*.

System and Access Requirements

- Oracle Integration, Version 23.02 or higher
- A secure FTP (sFTP) server or File Server
- An FTP client to access the external sFTP server

Before You Install the Recipe

Perform the following tasks on your sFTP server.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an input directory (FTP_COPY_LIST_FILES_READ) on the server from which to read files. The path of the directory should be: \FTP_COPY_LIST_FILES_READ\.
3. Place a sample file (test.xml) in the input directory.
4. Create an output directory (FTP_COPY_LIST_FILES_WRITE) on the server to write files. The path of the directory should be: \FTP_COPY_LIST_FILES_WRITE\.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle FTP Connection

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

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

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

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

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your FTP server and check if the list of files in the input directory have been copied to the output directory.

Related Documentation

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

Copy Objects from OCI Object Storage to Microsoft SharePoint

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

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

System and Access Requirements

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

Before You Install the Recipe

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

Configure OCI Object Storage

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

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

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

Configure SharePoint

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

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

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

- c. Click **Create**.

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

- d. Provide permission to the add-in that you created.
 - i. Go to: `https://<sharepoint_domain>.sharepoint.com/sites/<sharepointSiteName>/_layouts/15/appinv.aspx`
 - ii. Enter the client Id in the **App Id** field and click the **Lookup** button. The **Title**, **App Domain**, and **Redirect URL** fields will be populated automatically.
 - iii. Enter the following in the **App's permission Request XML** text area.

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

- iv. Click **Create**.
 - v. In the resulting window, click **Trust It**.

2. Get the SharePoint tenant Id.
 - a. Go to: `https://<sharepoint_domain>.sharepoint.com/_layouts/15/appprincipals.aspx`.

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

This is the SharePoint tenant Id.
3. Get the SharePoint resource Id.
 - a. Enter the following POST endpoint in Postman.

`https://<sharepoint_domain>.sharepoint.com/_vti_bin/client.svc/`
 - b. Add the **Bearer** type Authorization header without entering any bearer token and click **Send**.
 - c. In Postman, go to the **WWW-Authenticate** response header and copy the value of the client id.

This is the SharePoint resource Id.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the SharePoint REST Connection

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

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

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

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

Configure the Lookup Table

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

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

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

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

Activate and Run the Recipe

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

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

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

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

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

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Create Accounting Journals in Oracle Accounting Hub

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

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

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

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

System and Access Requirements

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

Before You Install the Recipe

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

Access Your FTP Server

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

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

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

Configure Oracle ATP

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

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

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

For more information, refer to [Oracle Autonomous Database](#) documentation.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign required roles to the user account.
 - a. Create a new job role under role category Common - Job Roles.
 - b. Under role hierarchy, add the following duties:
 - `ORA_XLA_ACCOUNTING_HUB_INTEGRATION_DUTY`
 - `ORA_XLA_ACCOUNTING_HUB_INTEGRATION_DUTY_OBI`
 - `FIN_FUSIONACCOUNTINGHUB_IMPORT_RWD`

- c. Save the job role.
- d. Assign the job role to a user.
- e. Run the following jobs:
 - Send Pending LDAP Requests
 - Retrieve Latest LDAP Changes
 - Import User and Role Application Security Data

Configure Oracle Accounting Hub

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

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

For more information, refer to the following:

- The **Set up Accounting Hub Cloud** section in this [blog](#).
- [Accounting Hub Best Practices](#).
- [Implementing Accounting Hub](#)

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ATP ImportAccountTransaction Connection

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

Field	Information to Enter
Security Policy	Select JDBC With OCI Signature .
Wallet	Select the check box, then click Upload to upload the wallet file. See Configure Oracle ATP .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.

Field	Information to Enter
Database Service Password	Enter the database service password to connect to the ATP database.
Object Storage Region	Specify the region in which your object storage is located. For example, us-ashburn-1.
Object Storage Tenancy OCID	Specify the value you copied from the Oracle Cloud Console. For example, <code>ocid1.tenancy.oc1.alphanumeric.value</code>
Object Storage Compartment OCID	Specify the value you copied from the Oracle Cloud Console. For example, <code>ocid1.compartment.oc1.alphanumeric.value</code>
Object Storage User OCID	Specify the value you copied from the Oracle Cloud Console. For example, <code>ocid1.user.oc1.alphanumeric.value</code>
Private Key	Click Upload to select the key you created. Ensure that the key is in RSA (PKCS1) format.

Note

Only a private key *without* a pass phrase/ password is supported.

Finger Print	Enter the finger print that was generated when you created the key in the Oracle Cloud Console.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle ERP SOAP ESS ImportAccountTrans Connection

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

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

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle FTP ImportAccountTransactions Connection

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

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

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

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

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

A message confirms if your test is successful.

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

Configure the Oracle ERP ImportAccountTransactions Connection

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

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

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

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

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

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

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

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

Related Documentation

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

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

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

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

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

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- HubSpot
- An account on HubSpot with the Administrator role
- Oracle CX Sales and B2B Service
- An account on Oracle CX Sales and B2B Service with the Administrator role

Before You Install the Recipe

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

Configure HubSpot

To configure HubSpot, create a company. See [Create companies](#).

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

Configure Oracle CX Sales and B2B Service

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

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

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

For the procedure to create the integration user account, see *Creating an Integration User Account in Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.

2. Assign the following roles to the user account:

- Integration Specialist
- AttachmentsUser
- FND_MANAGE_CATALOG_SERVICE_PRIV

For steps to assign roles, see *Assign Integration Roles in Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.

3. Make a note of the user name and password you set for the account.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle HubSpot Connection

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

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

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

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

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

You'll now be redirected to the HubSpot sign in page.

7. Enter your HubSpot account credentials.

A page is displayed asking for permission to interact with the account.

8. Click **Yes**.

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

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

A message confirms if your test is successful.

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

Configure the Oracle B2B Connection

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

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

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

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

Activate and Run the Recipe

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

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

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

- c. Click **Submit**.

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

You've now successfully submitted the integration for execution.

Note

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

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

Related Documentation

- [Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3](#)
- [Using the HubSpot Adapter with Oracle Integration 3](#)

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

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

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

System and Access Requirements

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

Before You Install the Recipe

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

Configure Oracle Field Service Cloud

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

To add an application:

1. Log in to an Oracle Field Service Cloud instance.
2. In the top-left corner, click the hamburger menu, then click **Configuration**.
3. On the Configuration page, under **Subsystems**, click **Applications**.
4. Click **Add Application**.

5. Enter the **Application Name** and **Application ID**, give the required API access. See [Authorize Access to APIs](#).
6. Click **Submit**.
7. Select the type of token service for the registered client application from the **Token service** drop-down list.

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

Configure Oracle ERP Cloud

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

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

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

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

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

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

For more information, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

3. Update the **Work Order Header** flexfield with two new segments, `senttoofsc` and `OFSC SYNC STATUS`.
 - a. In the top-right corner of the Oracle ERP Cloud home page, click the profile image, then select **Setup and Maintenance**.
 - b. On the right, click **Task**, and select **Search**.
 - c. In the Search field, enter `Manage Descriptive Flexfields`, and then click **Search**.
 - d. From the search results, select **Manage Descriptive Flexfields**.
 - e. On the Manage Descriptive Flexfields page, select **Work Orders** from the Module drop-down menu, then click **Search**.
 - f. From the search results, select **Work Order Header** and click **Edit**.
 - g. On the Edit Descriptive Flexfield page, click the **Create** icon in the Global Segments section. See [Create Global Segments](#).
 - h. On the Create Segment page, enter the name for the segment as `senttoofsc`, and enter other values for the segment as required.

- i. Repeat steps 3g and 3h to create another segment named `OFSC_SYNC_STATUS`.
 - j. Click **Save and Close**.
4. Deploy the descriptive flexfield. See [Deploy Descriptive Flexfields](#).

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Field Service Cloud Connection

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

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

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

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

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

A message confirms if your test is successful.

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

Configure the Oracle ERP Cloud Connection

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

Field	Information to Enter
Security Policy	Select Username Password Token .

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

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

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

Create a maintenance work order in Oracle ERP Cloud.

- a. Log in to your Oracle ERP Cloud instance.
- b. Under Supply Chain Execution, click **Maintenance Management**.
- c. Click the **Task** icon on the right and select **Manage Maintenance Work Orders**.
- d. On the Maintenance Work Orders page, click **Create**.
- e. In the Create Maintenance Work Order window, enter the details of the new work order and click **Save and Close**.

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

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

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check for the new activities created in your Oracle Field Service Cloud instance.
 - a. Log in to the Oracle Field Service Cloud instance.

- b. Select **Dispatch Console** in the left navigation pane.
- c. Check if a corresponding activity is created for the maintenance work order created in Oracle ERP Cloud.

Related Documentation

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

Create Activities in Oracle Primavera Cloud for Jira Issues

Use this recipe to create corresponding activities in Oracle Primavera Cloud for issues retrieved from Jira based on a filter ID.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates activities in Oracle Primavera Cloud for each issue retrieved from Jira based on a filter ID, as per a schedule specified in Oracle Integration.

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

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462) or higher
- Atlassian
- Oracle Primavera Cloud
- An account on Atlassian with the Administrator role
- An account on Oracle Primavera Cloud 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
- At least one project in your Oracle Primavera Cloud account

Before You Install the Recipe

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

Configure Jira

To access Jira from Oracle Integration, you must perform the following configuration tasks:

1. Create an app, configure the OAuth 2.0 authorization code, and assign the scopes. See *Prerequisites to Use the Authorization Code Credentials Security Policy in Using the Jira Adapter with Oracle Integration 3*.
2. Create a custom field in Jira to store the Oracle Primavera Cloud activity ID. See [Create a custom field](#). Specify the name of the field as PrimaveraActivity.
3. Note the ID of the custom field. See [Edit a custom field](#). When you click **Edit Details**, the Edit Custom Field Details page appears. From the URL displayed in the browser, note the ID available at the end of the URL. For instance, id=10064.
4. Create an issue filter in Jira and obtain its ID.
 - a. To create a filter, see [Save your search](#).
 - b. To view and obtain the filter ID, see [Managing your existing filters](#). Open the filter, and from the URL displayed in the browser, note the ID available at the end of the URL. For instance, filter=10001.

Configure Oracle Primavera Cloud

Create a custom field in your Oracle Primavera Cloud project. See [Add a Project Field](#) in *Oracle Primavera Cloud Schedule Management User Guide*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Jira Connection

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

Field	Information to Enter
Security Policy	Select Authorization Code Credentials .
Client ID	Enter the integration key of the OAuth application created in Jira. See Configure Jira .
Client Secret	Enter the client secret of the OAuth application created in Jira.

Field	Information to Enter
Scope	Enter read:jira-work, manage:jira-project, write:jira-work, manage:jira-configuration, read:jira-user.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle REST Primavera Connection

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

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter your Oracle Primavera Cloud instance URL.

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Enter https://<Primavera_Host_Name>/primediscovery/apitoken/request.
Client Id	Enter the username of your Oracle Primavera Cloud instance.
Client Secret	Enter the password of your Oracle Primavera Cloud instance.
Scope	Enter http://{Primavera_Host_Name}/api.
Client Authentication	Select Send client credentials as basic auth header .

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the following lookup tables to map the fields between your Jira and Oracle Primavera Cloud instances.

- ORACLE-BRT-PRIMAVERA_CONSTANTS**: Used to map the fields between Jira and Oracle Primavera Cloud.
- ORACLE-BRT-JIRA_PRIMAVERA_STATUS**: Used to map the status of the issue in Jira with the status of the corresponding activity in Oracle Primavera Cloud.

1. In the Lookups section, click the lookup name.
2. To configure the **ORACLE-BRT-PRIMAVERA_CONSTANTS** lookup table:
 - a. In the **Name** column, enter the name of the Oracle Primavera Cloud field you want to map.
 - b. In the **Value** column, enter the name of the field in Jira to which the Oracle Primavera Cloud field has to be mapped.
3. To configure the **ORACLE-BRT-JIRA_PRIMAVERA_STATUS** lookup table:
 - a. In the **JIRA_STATUS** column, enter the status of the issue in Jira.
 - b. In the **PRIMAVERA_STATUS** column, enter the status of the corresponding activity in Oracle Primavera Cloud.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Configure the Integration Flow

Update the integration flow of the recipe (**Oracle Jira Primavera Task Activity Sync**) with the ID of the custom field you created in Jira.

1. In the Integrations section, click the integration flow name.
2. Configure the Switch condition.
 - a. On the integration canvas, click the first switch case (**IF customfield_10064**), and then click **Actions**  and select **Edit**.
 - b. On the resulting page, select **Expression Mode**. In the Source section, search and select the ID of the custom field you created in Jira previously. See [Configure Jira](#).
 - c. Click the move button to add the custom field to the expression editor.
 - d. Click **Validate**, and then **Close**.
3. Update the data mapping.
 - a. On the integration canvas, click the **Map to updateActivity** element, and then click **Actions**  and select **Edit** to open the mapper.
 - b. In the data mapper, under Source section search for the ID of the custom field you created in Jira previously and map it to **Activity Code** within the **Request Wrapper**.
 - c. Click **Validate**, and then **Close**.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

In the first run, the recipe creates corresponding activities in Oracle Primavera Cloud for issues retrieved from Jira based on a filter ID. In subsequent runs, it creates activities for new Jira issues (that match the filter) and also propagates updates made in earlier Jira issues to the corresponding Oracle Primavera Cloud activities

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check for the new (or updated) activities.

Log in to your Oracle Primavera Cloud instance. From the menu, select **Schedule** and then select **Activities**.

Related Documentation

- [Using the Jira Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)

Create Advance Shipment Notices in Oracle Fusion Cloud SCM

Use this recipe to create an Advance Shipment Notice (ASN) in Oracle Fusion Cloud Supply Chain & Manufacturing (SCM).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to create advance shipment notices in Oracle Fusion Cloud SCM. An Advance Shipment Notice (ASN) informs the receiving organization about an upcoming shipment, and it can be transmitted in various formats such as Electronic Data Interchange (EDI) or JSON.

This recipe is designed for suppliers to enable them to submit the necessary information in JSON format using the REST-based interface. Upon receipt of the request, the interface will generate an ASN in Oracle Fusion Cloud SCM and send a response message back to the supplier confirming the ASN creation. If any issues arise during ASN creation, the status along with an error message will be returned to inform the requester of the problem.

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

System and Access Requirements

- Oracle Integration 3

- Oracle Fusion Cloud SCM
- Oracle Fusion Cloud SCM credentials

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance to successfully connect to this external system using Oracle Integration and create an Advance Shipment Notice (ASN) in Oracle Fusion Cloud SCM.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the ERP Connection

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

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

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

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. To create an Advanced Shipment Notice (ASN) from an external application, send a POST request to this endpoint URL along with the required payload.
 - e. The recipe creates the Advanced Shipment Notice in Oracle Fusion Cloud SCM and provides the response.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **CREATE ASN to Oracle SCM** integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click the **Body** tab and enter the message as the request payload.

Example of a request payload to create ASN:

```
{
  "ReceiptSourceCode":"VENDOR",
  "ASNTYPE":"ASN", // It will be hardcoded in case ASN
  "OrganizationCode":"001",
  "VendorName":"Advanced Corp", // Supplier Name
  "VendorSiteCode":"AC US1", // Supplier Site Code
  "ShipmentNumber": "ASN-12-0987661", // ASN Number to create.
  "ShippedDate": "2024-09-09",
  "lines":[
    {
      "ReceiptSourceCode":"VENDOR",
      "TransactionType":"SHIP",
      "AutoTransactCode":"SHIP",
      "SourceDocumentCode":"PO",
      "OrganizationCode":"001",
      "DocumentNumber":"US1641571", // Put a valid PO Number
      "DocumentLineNumber":2, // Enter the correct Line# against the
      ItemNumber entered
      in ItemNumber column.
      "ItemNumber":"AS65004", //Take correct item from PO item# column else
      ASN would
      not be created
    }
  ]
}
```

```
"Quantity":1,
"UnitOfMeasure":"Ea",
"SoldtoLegalEntity":"US1 Legal Entity"
}
]
```

Success response if ASN is created:

```
{
  "Status" : "SUCCESS",
  "ASNType" : "ASN",
  "ShipmentNumber" : "ASN-12-0987661",
  "CreatedBy" : "AAA.TTT",
  "CreationDate" : "2024-09-12T07:14:30.001+00:00"
}
```

Failure response if ASN is not created:

```
{
  "Status" : "ERROR",
  "Error" : "RCV_INVALID_ROI_VALUE_NE",
  "ErrorMessage" : "You must enter a valid value in the LINE_NUM column.
The current value
is 2."
}
```

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Verify that the ASN is created in Oracle Fusion Cloud SCM.
 - a. Log in to Oracle Fusion Cloud SCM.
 - b. From the navigation menu on the left, expand **Supply Chain Execution** and click **Inventory Management**.
 - c. On the page that displays, click the **Search** icon.
 - d. From the opened drawer, choose **Expected Shipment Line** from the first drop-down list and then choose **ASN** from the second drop-down list.
 - e. Enter the ASN number (**ShipmentNumber**) which is passed in the request payload. If the ASN is created successfully, it will show the ASN number against the Purchase Order (**DocumentNumber**) that is given in the request payload.

Related Documentation

- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)

Create Alerts in Microsoft Teams for Jira Issues

Use this recipe to create an alert in Microsoft Teams whenever an issue is created in Jira.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)

- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that sends an alert in Microsoft Teams whenever an issue is created in Jira.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. The **JIRA MS Teams Issue Alert** integration flow subscribes to the issue create event in Jira. Whenever a new issue is created, the integration flow of the recipe is triggered, and it creates a corresponding alert in Microsoft Teams. Basic data is synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 23.06.2.0.0 or higher
- Atlassian
- An account on Atlassian with the Administrator role
- At least one Jira project on your Atlassian account. 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
- Microsoft Teams
- An account on Microsoft Teams with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Atlassian and Microsoft Teams instances in order to successfully connect to these external systems using Oracle Integration and create alerts.

Configure Jira

To access Jira from Oracle Integration, you must perform the following configuration tasks:

1. Create an app, configure the OAuth 2.0 authorization code, and assign the scopes. See [Prerequisites to Use the Authorization Code Credentials Security Policy](#) in *Using the Jira Adapter with Oracle Integration 3*.
 - In the Callback URL field, specify the callback URL in the following format: `https://{OIC_Hostname}:443/icsapis/agent/oauth/callback`
 - Ensure that you add the following scopes: `read:jira-work` `manage:jira-project` `write:jira-work` `manage:jira-configuration` `read:jira-user`
2. Create a webhook. See [Registering a webhook via the Jira administration console](#) in Atlassian Developer documentation.
 - In the URL field, enter your Oracle Integration endpoint URL. For example, `https://<encodedUsername>:<encodedPassword>@<host>/ic/api/integration/v2/flows/jira/project/ORCL-R-JIRA_MSTEAMS_ISSUE_ALERT/JIRA_MS_TEAMS_ISSUE_ALERT/1.0/Inbound/issue_created?user_id=[user_id-value]&user_key=[user_key-value]`.
 - In the Events field, select **Issue Created**.

Configure Microsoft Teams

Complete certain configuration tasks in your Microsoft Teams 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, Jira-MicrosoftTeams.
6. In the Supported account types section, select who can use this application or access this API.
7. 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 Teams 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:
 - Enter a description in the **Description** field.
 - Select the expiration time of the app from the options available in the **Expires** field.
 - 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 Teams connection from Oracle Integration.
11. Set the permissions and scopes for the application.
 - a. In the left navigation pane, click **API permissions**, and select **Microsoft Graph**.
 - b. In the API permissions page, select **delegated permissions**. Click **Add a permission**, and add the following API permissions to the application.
 - `ChannelMessage.Send`
 - `Chat.ReadWrite`
 - `ChatMessage.Send`

- `Group.ReadWrite.All`

12. Navigate to the Overview page and click **Endpoints**. Note the OAuth2.0 authorization endpoint(v2) and OAuth 2.0 token endpoint(v2).

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Jira Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Jira host name. For example: `https://your_domain_name.atlassian.net`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Authorization Code Credentials selected.
Client ID	Enter the Client ID obtained when you configured your Atlassian instance. See Configure Jira .
Client Secret	Enter the Client Secret obtained when you configured your Atlassian instance.

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

A message confirms if your test is successful.

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

Configure the Oracle REST MS Teams Connection

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

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter your Microsoft Graph URL. For example: <code>https://graph.microsoft.com/v1.0</code> .

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

Field	Information to Enter
Security Policy	Leave OAuth Authorization Code Credentials selected.
Client Id	Enter the Client ID obtained when you configured Microsoft Teams. See Configure Microsoft Teams .

Field	Information to Enter
Client Secret	Enter the Client Secret obtained when you configured Microsoft Teams.
Authorization Code URI	Enter the Authorization Code URI obtained when you configured Microsoft Teams.
Access Token URI	Enter the Access Token URI obtained when you configured Microsoft Teams.
Scope	Enter the following scope: <code>https://graph.microsoft.com/ChannelMessage.Send</code> <code>https://graph.microsoft.com/Chat.ReadWrite</code> <code>https://graph.microsoft.com/ChatMessage.Send</code> <code>offline_access</code>

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **JIR-MSTeams-Config-Lookup** lookup table to configure the fields in Microsoft Teams.

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

Key	Description
team-id	Specify the unique ID of your team in the Microsoft Teams instance.
channel-id	Specify the unique ID of your channel in the Microsoft Teams instance.
to-address	Specify the email address to which the error notification emails are sent. Enter an email address of your choice.
from-address	Specify the email address from which the error notification emails are sent. Enter an email address of your choice. See Troubleshoot Oracle Cloud Infrastructure Notification Email Configuration to Ensure Proper Delivery .

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

Activate and Run the Recipe

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

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the recipe. To do so, create an issue in Jira. See [Create an issue and a subtask](#) in Atlassian documentation.
You've now successfully triggered the recipe.
- Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

4. Check if you have received an alert in your Microsoft Teams instance.

Related Documentation

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

Create and Add a User to Multiple Groups in OCI

Use this recipe to create a user in Oracle Cloud Infrastructure (OCI) and add the user to multiple groups. The groups are specified as an array, and the newly created user will be added to all the existing groups from the array.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe offers an efficient solution to manage user creation and group assignments in Oracle Cloud Infrastructure (OCI).

This recipe uses the user details and an array of group names as the input to automatically create a user and add the newly created user to all the existing groups specified in the array. This automation simplifies user provisioning, reduces manual effort, and ensures consistency in group membership, improving the overall efficiency of access management.

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

System and Access Requirements

- Oracle Integration 3
- Oracle Cloud Infrastructure domain access

Before You Install the Recipe

You must complete the following prerequisites before installing this recipe.

To call the Oracle Cloud Infrastructure APIs, the OCI Signature Version 1 security policy is used in the REST Adapter. To create the signing key, and obtain the tenancy OCID and user OCID, see the steps documented in OCI Signature Version 1 Security Policy Use in *Using the REST Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the OCI Domain REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the connection URL. For example: `https://domain_URL`. See [Finding an Identity Domain URL](#) to locate the domain URL you need.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OCI Signature Version 1 .
Tenancy OCID	Enter the Oracle Cloud Infrastructure (OCI) tenancy OCID. See Before You Install the Recipe .
User OCID	Enter the OCI user OCID. See Before You Install the Recipe .
Private Key	Upload the private key. See Before You Install the Recipe .
Fingerprint	Enter the fingerprint. See Before You Install the Recipe .

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. To create a user from an external application, send a POST request to this endpoint URL along with the required payload.
 - e. The recipe creates a user and assigns the user to groups in the Oracle Cloud Infrastructure (OCI) domain.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Create User and Add in Groups** integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click the **Body** tab and enter the message as the request payload.

Example of a request payload to create a user:

```
{
  "lastName": "Doe",
  "firstName": "John",
  "groups": ["group1", "group2"],
  "userName": "jd",
  "email": "j.d@example.com"
}
```

Success response if user is created:

```
{
  "status" : "Success",
  "displayName" : "John Doe",
  "id" : "850f36f655374b3c81d95ba9b0dd8a20",
  "firstName" : "John",
  "lastName" : "Doe",
  "email" : "j.d@example.com",
  "userName" : "jd"
}
```

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Verify that the user is created in Oracle Cloud Infrastructure (OCI) under Identity Domain.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- [OCI Identity Domains API documentation](#)

Create Box Folders for ServiceNow Incidents

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

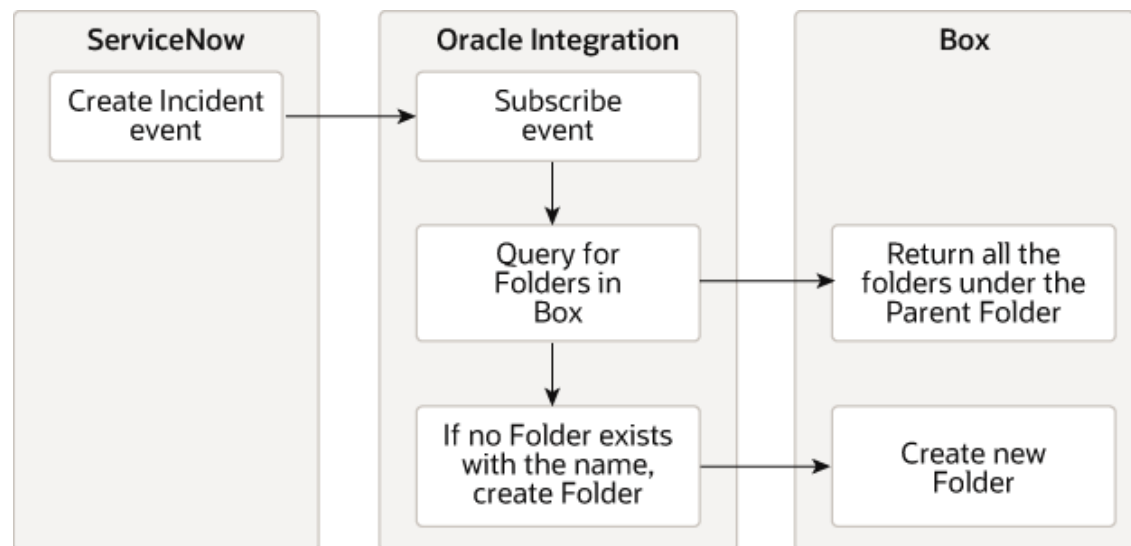
System and Access Requirements

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



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.

Configure ServiceNow

To access ServiceNow using Oracle Integration and create corresponding Box folders for ServiceNow incidents, you must perform certain configurations on your ServiceNow instance.

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

1. Create a custom user (for example, Integration user) and assign a custom role that has access to ServiceNow tables. See **Satisfy User and Role Requirements** under Prerequisites for Creating a Connection in *Using the ServiceNow Adapter with Oracle Integration 3*.
2. Enable web services and assign permissions to required tables.
 - a. On the ServiceNow home page, search for tables in the search box in the left pane, and click the **Tables** link under System Definition in the search results.
 - b. Search for each of the following ServiceNow tables using the Search box or locate a table using the show/hide filter.
 - sys_soap_message
 - sys_soap_message_function
 - sys_script
 - c. Click the table name in the search results.
 - d. Locate and click the **Application Access** tab.
 - e. Select the respective permission (refer to the following table for required permissions), and select the **Allow access to this table via web services** check box if it is not selected already.

Table Name	Permission
sys_soap_message	Read, Create, Update, Delete
sys_soap_message_function	Read, Create, Update, Delete
sys_script	Read, Create, Update, Delete

3. Enable multiple insert plugin to support insertMultiple in SOAP.
 - a. On the ServiceNow home page, search for **System Definition**.
 - b. Select **Plugins**.
 - c. Locate and click **Insert Multiple Web Service**.
 - d. If the status is inactive, click **Activate/Repair**.
4. Get the ServiceNow Caller ID.
 - a. On the ServiceNow home page, search for **User Administration** and then click **Users**.
 - b. On the Users page, select the custom user that you configured earlier in step 1.
 - c. Locate the value of the sys_id in the URL that appears when you select the user.
 - d. Note the sys_id value as you would need to provide it while configuring lookup table in Oracle Integration.

In addition, 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**.
4. 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.
8. 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.
 - a. Log in to the Box Developer console at <https://developer.box.com>.
 - b. Create a custom app.
 - i. Click **Create New App**.
 - ii. In the resulting page to choose the application type, select **Custom App**.
 - iii. In the Custom App dialog, select the option **User Authentication (OAuth 2.0)**.
 - iv. Enter a unique name for your app in the **App Name** field.
 - v. Click **Create App**.

The Box app's configuration page appears.
 - c. Get the OAuth 2.0 credentials.
 - i. In the Box app's configuration page, scroll to the **OAuth 2.0 Credentials** section.
 - ii. Copy the client ID from the **Client ID** field, and the client secret from the **Client Secret** field.
 - iii. Note the Client ID and Secret. You'll need these later while configuring the Box connection from Oracle Integration.
 - d. Specify the redirect URL.
 - i. In the Box app's configuration page, scroll to the **OAuth 2.0 Redirect URI** section.
 - ii. 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.

- e. Get the Signature Keys.
 - i. Click the **Webhooks** tab on the top of the page.
 - ii. Click **Generate Key** to generate the primary and the secondary keys.
 - iii. Copy the primary and the secondary keys.
 - iv. 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 Your Signature Keys in *Using the Box Adapter with Oracle Integration 3*.

2. 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.

- a. Log in to your Box account at <https://account.box.com/login>.
- b. Click **New** on the top right side of the All Files page.
- c. From the menu that displays, click **Folder**.
- d. In the Create a New Folder dialog, enter a suitable name in the **Folder Name** field.
- e. 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.

- f. Click **Create**.

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

3. 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.

- a. On the All Files page of your Box instance, search and locate the parent folder that you created.
- b. Hover your mouse on the parent folder.
- c. Note the link that appears below on the left side of the page. For example, `https://app.box.com/folder/135666632045`.
- d. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ServiceNow Connection

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

Field	Information to Enter
Username	Enter your ServiceNow username.
Password	Enter your ServiceNow password.

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

A message confirms if your test is successful.

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

Configure the Oracle Box Connection

1. In the Connections section, click the connection name.
2. 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 [Configure Box](#).

3. 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.

4. 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.
5. Click **Provide Consent**.
6. 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.

7. Enter your Box account credentials, and click **Authorize**.
8. 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.

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

A message confirms if your test is successful.

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

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.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values. The recipe's integration flow contains the following properties:
 - a. **NotificationMail**: This integration property holds the email address to which error notification emails are sent. Enter an email address of your choice.
 - b. **ParentFolderID**: This integration property holds the ID of the parent folder that you created in Box. See [Configure Box](#).
3. Click **Submit**.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Configure the basic authentication in ServiceNow for the integration flow of the recipe.
 - a. After activating the recipe, obtain the metadata URL of the integration flow.
 - i. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run details**.
 - ii. In the Run details panel, copy the **Metadata URL** value.
 - b. Use the metadata URL to configure a basic authentication in ServiceNow. This is necessary to invoke the integration flow in Oracle Integration from ServiceNow.

See Invoke an Integration from ServiceNow in *Using the ServiceNow Adapter with Oracle Integration 3*.

3. Run the recipe.

Create an incident in ServiceNow.

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

The Self-Service menu displays all related modules.
- c. In the Self-Service menu, click **Incidents**.
- d. In the Incidents window, click **New**.
- e. Fill the required information 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.

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. 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 3*
- *Using the Box Adapter with Oracle Integration 3*

Create Box Folders for Zendesk Organizations

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

Note

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
- An account on Zendesk with the Administrator role
- Box
- An account on Box with the Administrator role

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

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.

1. 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 its Configuration page is displayed.
 - e. On the application's Configuration page:
 - i. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Zendesk Connection

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

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

3. 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.

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

A message confirms if your test is successful.

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

Configure the Oracle Box Connection

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

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

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

You'll now be redirected to `box.com`, where you are prompted to authorize the connection to the Box OAuth application created previously.

6. 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.

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

A message confirms if your test is successful.

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

Configure the 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. In the Lookups section, click the lookup name.
2. Against the **ParentFolderId** field, enter the ID of the parent folder in Box. See [Configure Box](#).
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**. Note that the **Ad hoc request** radio button is selected by default.

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

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your 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 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)

Create Customer Records in Oracle ERP Cloud

Use this recipe to create or update a customer record in Oracle ERP Cloud through a REST request from an external application.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates or updates a customer record along with account and profile details in Oracle ERP Cloud when it receives a REST request with the customer data from an external application. It contains an app-driven orchestration integration, which uses the standard REST Adapter and the Oracle ERP Cloud Adapter.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate the integration flow and send a REST request with the customer data to the integration's endpoint URL from an external application. When triggered, the integration flow checks if the customer already exists based on the `SourceSystemReferenceID` field. If the customer exists, the recipe updates the customer record, or else it creates a new customer record and corresponding account, and customer profile based on the request payload data. The customer profile is created if the `customerProfileType` field exists in the payload.

Note

- If the account creation fails after the customer record has been created, you can use the same payload in the subsequent runs to update the customer record with the account details.
- You can also create multiple accounts for a customer record. If a customer record has an account already associated with it, the incoming payload is checked. If the `customerAccountName` field in the payload is different from the existing account, a new account is created for the customer record.

System and Access Requirements

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

Assumptions

- The Oracle ERP Cloud instance has Spoke system for the source system. See [Product Spoke Systems](#).

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance in order to successfully connect to these external systems using Oracle Integration and create customer records.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist

- Oracle ERP Cloud-specific data access to the integration user

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ERP Cloud Connection

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

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

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

A message confirms if your test is successful.

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

Configure the Oracle REST Trigger Connection

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the integration flow. You can update the integration properties only after activating this flow.

- a. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
- b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
SetId	Specify the SetID of the organization. SetIDs enable you to share a set of reference data across organizations. For example, 300000000002169.
Party_Usage_Code	Specify the party usage code that illustrates the usage of parties across applications in business flows. For example, EXTERNAL_LEGAL_ENTITY.
Created_By_Module	Specify the module code. For example, HZ_WS.

- c. Click **Submit**.

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

3. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the 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 ERP Cloud and returns the internal ID of the newly-created record as a response.

4. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click **Body** and enter sample customer data. For example:

```
{
  "sourceSystem": "VBApp",
  "sourceSystemCustomerID": "Acme Corporation",
  "customerName": "Acme Consulting",
  "customerEmail": "contact@acmecorp.com",
  "customerPhone": "042400000",
  "customerAccountName": "SampleAccountName",
  "customerAddress": [
    {
      "AddressLine": "addr1",
      "City": "Sydney",
      "Postcode": "2102",
      "State": "NSW",
      "Country": "AU",
      "Type": "BillTo"
    }
  ],
}
```

```

        "AddressLine": "addr2",
        "City": "Sydney",
        "Postcode": "2102",
        "State": "NSW",
        "Country": "AU",
        "Type": "ShipTo"
    },
    ],
    "customerProfile": {
        "customerProfileType": "DEFAULT",
        "CreditChecking": "Y",
        "CreditHold": "N",
        "DiscountTerms": "N",
        "EffectiveEndDate": "",
        "EffectiveStartDate": "",
        "AutoReceiptsIncludeDisputedItems": "",
        "ConsolidatedInvoice": "",
        "CreditLimit": "1000",
        "CreditCurrencyCode": "USD",
        "CollectorName": "James"
    }
}

```

c. Click **Run**

The integration now creates the customer record in Oracle ERP Cloud. The Activity Stream pane appears displaying the status of the integration instance's execution.

d. 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 Oracle ERP Cloud.

5. Log in to your Oracle ERP Cloud instance as an Administrator and check for the new customer record.

Navigate to **Billing Work Area**, select **Manage Customers**, and search for the specific customer record.

Related Documentation

- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)

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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 and configure the connections and other resources within the recipe. Subsequently, you can activate the integration in the recipe 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.

Note

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

Before You Install the Recipe

To successfully connect to Oracle NetSuite using Oracle Integration and create customer records, you must perform certain general configurations on your Oracle 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 Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See [Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).
2. Perform the TBA-related configuration tasks. See [Prerequisites for the Token-Based Authentication Security Policy in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Lists	Currency	Full
Lists	Customers	Full
Lists	Subsidiaries	Full
Lists	Contacts	Full
Setup	SOAP Web Services	Full
Setup	User Access Tokens	Full

Tab Name	Permission	Level
Setup	Log in using Access Tokens	Full

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle REST Connection

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

A message confirms if your test is successful.

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

Configure the Oracle NetSuite Connection

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

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, <code>https://webservices.netsuite.com/wsd/ <NetSuite_application_version>/netsuite.wsdl..</code> where <code><NetSuite_application_version></code> is the version of the Oracle NetSuite application. For example, <code>v2015_1_0</code> . See Assemble the Oracle NetSuite WSDL URL in <i>Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

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

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Create an Integration Record for Oracle Integration in <i>Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See Create an Access Token for the User Account in <i>Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Token Secret	Enter the token secret provided by Oracle NetSuite.

Field	Information to Enter
Account ID	Enter your Oracle NetSuite account identifier. See Make a Note of the NetSuite Account ID in <i>Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

Note

You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the 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.

3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click **Body** and enter 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"
}
```

c. Click **Run**

The integration now creates the customer record in Oracle NetSuite. The Activity Stream pane appears displaying the status of the integration instance's execution.

d. 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 Oracle NetSuite.

4. Log in to your Oracle NetSuite instance as an Administrator and check for the new customer record.

a. On the Oracle 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.

c. 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 3*
- *Using the REST Adapter with Oracle Integration 3*

Create Customers in Magento Commerce Cloud for Oracle ERP Cloud Accounts

Use this recipe to create customers in Magento Commerce Cloud for Oracle ERP Cloud accounts.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a customer record in Magento Commerce Cloud whenever an account is created in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. The recipe uses the Oracle ERP Cloud Adapter as a subscriber to **Account Create** and **Account Update** events. When a new account is created in Oracle ERP Cloud, the **Oracle ERP Magento Account Create** integration flow of the recipe is triggered, and it creates a corresponding customer record in Magento Commerce Cloud.

Further, whenever an account is updated in Oracle ERP Cloud, the **Oracle ERP Magento Account Update** integration flow of the recipe is triggered, and it checks the custom field of the account record that stores the Magento ID. If the custom field is empty, the integration flow creates a new customer record in Magento Commerce Cloud. If not, it updates the existing customer record. Basic data such as customer name, billing city, billing country, and postal

code are synchronized between the Oracle ERP Cloud and Magento Commerce Cloud platforms.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- Magento Commerce Cloud
- An account on Magento Commerce Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Magento Commerce Cloud instances in order to successfully connect to these external systems using Oracle Integration and create customers in Magento Commerce Cloud for Oracle ERP Cloud accounts.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user
3. Create custom fields in Application Composer.
 - a. In the top-left corner of the Oracle ERP Cloud home page, click **Menu**, expand **Configuration**, and select **Sandboxes**.
 - b. In the Sandboxes page, click **Create Sandbox**.
 - c. In the Create Sandbox page, specify the name of the Sandbox, and from the All Tools section, select **Application Composer**. Further, click **Create and Enter**.
 - d. In the top-left corner of the Oracle ERP Cloud home page, click **Menu**, expand **Tools**, and select **Application Composer**.
 - e. In the Application Composer page, expand **Standard Objects**, and expand **Account**, and then select **Fields**.
 - f. In the Fields page, click **Create a custom field**.
 - g. In the Select Field Type pop-up, select **Text**, and click **OK**.
 - h. In the Create Text Field page, specify the name of the field as `MagentoCustomerId`, and click **Save and Close**.

Similarly, repeat steps 3d. to 3h. to create custom fields for `MagentoEmailId`, `MagentoAccountLastName`, and `MagentoPhoneNumber`.

4. Add the custom fields to the page layouts.
 - a. In the Application Composer page, select **Pages**, and select **Creation Page Layouts**. Further, select **Default Custom Layouts**.
 - b. Click **Edit** and add all the custom fields created in the previous step.
 - c. Click **Save**.

Similarly, repeat step 4 to add all the custom fields created in the previous step to the **Details Page Layouts**.

5. Publish the changes.

In the top-left corner, select **MagentoFields**, and select **Publish**.

Configure Magento Commerce Cloud

To access the Magento Commerce Cloud instance from Oracle Integration and create customers, you'll need to create a user account for Oracle Integration on Magento Commerce Cloud and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Magento Commerce Cloud Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ERP Cloud Connection

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

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

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

A message confirms if your test is successful.

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

Configure the Oracle Magento Commerce Cloud Connection

Use the following steps to configure the **Oracle Magento Commerce Cloud Connection** connection.

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

Field	Information to Enter
Hostname	Specify the instance URL of your Magento Commerce Cloud instance.

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

Field	Information to Enter
Consumer Key	Enter the consumer key obtained when you configured Magento Commerce Cloud. See Configure Magento Commerce Cloud .
Consumer Secret	Enter the consumer secret obtained when you configured Magento Commerce Cloud.
Access Token	Enter the access token obtained when you configured Magento Commerce Cloud.
Access Token Secret	Enter the access token secret obtained when you configured Magento Commerce Cloud.

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

A message confirms if your test is successful.

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

Update Integration Properties

Update the integration properties for the **Oracle ERP Magento Account Create** integration flow with appropriate values.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the following integration properties with appropriate values.
 - **FromAddress**: This integration property holds the email address from which notification emails are sent. Enter an email address of your choice.
 - **ToAddress**: This integration property holds the email address to which notification emails are sent. Enter an email address of your choice.
 - **StoreId**: This integration property holds the internal ID of the Magento Commerce Cloud store in which the customers must be created.
 - **RegionId**: This integration property holds the internal ID of the Magento Commerce Cloud region in which the customers must be created.
3. Click **Submit**.

Note

This is an app-driven integration, and you can update the property values only before activating the recipe.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe. To do so, create an account in Oracle ERP Cloud.
 - a. Sign in to Oracle ERP Cloud.
 - b. Navigate to the **Service** work area and click **Accounts**.
 - c. Click **Create Account**.
 - i. Enter the **Name**.
 - ii. Select **Customer** from the **Type** menu.
 - iii. Specify the account key in the appropriate field.
 - iv. Click **Save and Close**.

Note

The field in which you enter the account key depends on the value assigned to the SVC_CSS_ACCT_KEY_FIELD profile option. The account key should be specified in the attribute that you defined for the account key. See [Create a Customer Account in Oracle Fusion Service](#).

You've now successfully triggered the recipe.

When you create an account in Oracle ERP Cloud, the **Oracle ERP Magento Account Create** integration flow gets triggered and a corresponding customer record is created in Magento Commerce Cloud.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Magento Commerce Cloud instance and check for new customers. In the Magento Commerce Cloud home page, navigate to **Customers**, and select **All Customers**.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Magento Commerce Cloud Adapter with Oracle Integration 3*

Create Customers in Shopify for Salesforce Accounts

Use this recipe to create customer records in Shopify for accounts in Salesforce using Azure Service Bus.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application driven recipe and is triggered whenever an account is created in Salesforce. The account details are first published to Azure Service Bus. Subsequently a corresponding customer record is created in Shopify by subscribing to Azure Service Bus.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When an account is created in Salesforce, the integration flow is triggered. The recipe first publishes the account details to Azure Service Bus and subsequently it subscribes to Azure Service Bus and creates a corresponding customer record for the account in Shopify. Basic information such as ACCOUNT_NAME, ACCOUNT_EMAIL, ACCOUNT_PHONE, BILLINGSTREET, BILLINGCITY, BILLINGSTATE, BILLINGCOUNTRY AND BILLINGPOSTALCODE are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Shopify
- An account on Shopify with the Administrator role
- Azure Service Bus
- An account on Azure Service Bus with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce, Shopify and Azure Service Bus instances to successfully connect to these external systems using Oracle Integration and create corresponding customer records in Shopify for Salesforce accounts, using Azure Service Bus

Configure Salesforce

To access Salesforce from Oracle Integration and create customers for accounts, you must perform certain configurations 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.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an account to trigger the **Oracle Salesforce AzureSB Shopify** integration flow of the recipe. To do so, login to your Salesforce instance as an **Administrator** and perform the following steps:

1. You need username and password to configure connection with Oracle Integration. To retrieve the credentials, perform the tasks in Prerequisites to Use the Salesforce Username Password Security Policy in *Using the Salesforce Adapter with Oracle Integration 3*.
2. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
3. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Define your **Platform Event** which enables Salesforce adapter to receive account creation notification. See Prerequisites to Use Platform Events in *Using the Salesforce Adapter with Oracle Integration 3*.
5. Follow the steps below to create the **Apex Trigger**.
 - a. Log in to your Salesforce instance.
 - b. Go to profile name, and then **Setup**.
 - c. Search Apex Trigger in the **QuickFind** search box present on the left panel.
 - d. Click **Developer Console**.
 - e. In the Developer Console window that appears, click **File**, then **New**, and then **Apex Trigger**.
 - f. In the New Apex Trigger window, enter the name of the trigger in the **Name** field and select the **sObject** from the drop-down list.
 - g. Click **Submit**.

A new Apex trigger is created.
 - h. Write the business logic as per the requirement and save it. Debug any errors that you encounter.

The newly created trigger now appears in the Apex Trigger list.

Note

For this recipe, refer to the following sample Apex Trigger file:
[SFDC_SHOPIFY_CREATE_ACC_Trig_APEXCode.tgr](#).

Configure Azure Service Bus

To access Azure Service Bus from Oracle Integration you must perform certain configurations on the Azure Portal.

1. Register an application in the Azure portal and obtain the tenant ID and client ID. See *Register an Application in Using the Azure Service Bus Adapter with Oracle Integration 3*.
2. Create a new client secret. See *Create a New Client Secret in Using the Azure Service Bus Adapter with Oracle Integration 3*.
3. Create a new service bus namespace. See [Create a namespace in the Azure portal](#).
4. Ensure firewall port permissions are verified by:
 - a. Confirming accessibility to the Advanced Message Queuing Protocol (AMQP) ports 5671 and 5672.
 - b. Confirming that the endpoint is permitted through the firewall.
5. Create a queue within the namespace which you created in step3 above. For example: queue_sf_to_shopify.

You can now use this queue to consume and publish messages using Oracle Integration.

Configure Shopify

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

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Salesforce Connection

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

Field	Information to Enter
Select Salesforce.com Instance Type	Select Sandbox as the instance type. See Configure Salesforce .
API version	Enter 61 as the Salesforce API version.

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

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

Field	Information to Enter
Password	Enter the password of the account created for Oracle Integration on the Salesforce instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle Azure Service Bus Connection

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

Field	Information to Enter
Tenant Id	Enter the tenant Id which you receive after registering your app on Azure Portal. See Configure Azure Service Bus .
Namespace	Enter OICStandardNamespace as your Azure Service Bus Namespace.

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

Field	Information to Enter
Client Id	Enter the client Id. See Configure Azure Service Bus .
Client Secret	Enter the client secret.
Password	Enter the password of the account created for Oracle Integration on Azure Service Bus.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle Shopify Connection

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

Field	Information to Enter
Hostname	Enter the Shopify host name. The Shopify hostname appears in the Shopify home URL. For example: <code>https://hostname.myshopify.com</code>
Shopify-REST API version	Enter the REST API version. For example, 2024-01.

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

Field	Information to Enter
Security Policy	Select Shopify Security Policy .
Username	Enter the API Key you receive after performing the prerequisite step to create a connection. See Configure Shopify .
Password	Enter the password you receive after performing the prerequisite step to create a connection.
(optional)Shared Secret	Enter the shared secret you receive after performing the prerequisite step to create a connection.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Update Integration Properties

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **EmailTo**: This integration property holds the email id to which error notifications have to be sent.
 - b. **retryCount**: This integration property holds the number of retries when an invoke fails. The default value is 3.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Create an account in Salesforce.
 - i. Log in to your Salesforce instance.
 - ii. Go to the **Accounts** tab. A list of recently created accounts appear.
 - iii. Click **New**.
 - iv. In the Account Edit page, enter the following details
 - Account information such as **Account Name**, **Phone**, **email**, **City_Name**.
 - Address information
 - Any additional details, if required.
 - v. Click **Save**.

You've successfully created an account in Salesforce and triggered the **Oracle Salesforce Azure SB Receive Account Created** integration flow. The integration now creates a corresponding entry in Azure Service Bus queue for the newly created Salesforce account.

- b. View the newly created customer account in Azure Service Bus Queue.
 - i. Log in to your Azure Service Bus instance.
 - ii. Go to the namespace **OICStandardNamespace**.
 - iii. Refresh the queue list to fetch the message.
 - iv. Select the queue (for example, **queue_sf_to_shopify**) which you have created while configuring Azure Service Bus.
 - v. Click **Service Bus Explorer** then **Peek from Start** to see the complete message (here customer account details).
- c. Run the **Oracle AzureSB Shopify Customer Creation** integration flow.
 - i. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - ii. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The integration now fetches the account details from Azure Service Bus and creates a corresponding customer in Shopify.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if a new customer (corresponding to the new Salesforce account created in step 2a) has been created in Shopify.
 - a. Log in to your Shopify instance.
 - b. Go to **Customers**.
 - c. Refresh the page to see the newly created customer in the list.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the Azure Service Bus Adapter with Oracle Integration 3*

Create Invoices in Oracle ERP Cloud Using OCI Document Understanding AI Service

Use this recipe to create invoices in Oracle ERP Cloud using the artificial intelligence capabilities of Oracle Cloud Infrastructure Document Understanding (OCI Document Understanding).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe uses the artificial intelligence capabilities of OCI Document Understanding to automatically extract and process information from invoices received as PDF file attachments in Microsoft Office 365 Outlook and create corresponding invoices in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connection and other resources within it. The main integration flow runs on a schedule. It fetches invoice PDF files that are received as attachments from a Microsoft Office 365 Outlook inbox and then encodes the PDF files to base64 in Oracle Integration. Subsequently, it invokes the child integration. The child integration flow sends the base64 encoded files to OCI Document Understanding. Using the artificial intelligence capabilities of OCI Document Understanding, key value data is extracted from the files and the results are stored in JSON format. Finally, corresponding invoices are created in Oracle ERP Cloud via REST API calls.

Note that to successfully run this recipe, the input PDF files that you receive in Outlook have to be in a certain format. You can download the following file for reference: [Sample file](#).

System and Access Requirements

- Oracle Integration, 3
- OCI Document Understanding
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- Microsoft Azure
- An account on Microsoft Azure with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Microsoft and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration and create invoices in Oracle ERP Cloud.

Additionally, to use the OCI Document Understanding AI service to create invoices in Oracle ERP Cloud, you must complete the prerequisites described in Prerequisites for OCI AI services in *Using Integrations in Oracle Integration 3*.

Configure Microsoft Outlook

To configure Microsoft Office 365 Outlook, see Prerequisites for Creating a Connection in *Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign required roles to the user account. For more information on the roles, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Microsoft Outlook Connection

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

Field	Information to Enter
Client Id	Enter the client Id.
Client Secret	Enter the client secret.
Scope	Enter the scope URL. For example <code>https://graph.microsoft.com/Mail.ReadWrite</code> <code>https://graph.microsoft.com/Mail.Send offline_access</code>

3. Click **Provide Consent**. This enables Oracle Integration to interact with the Microsoft Office 365 Outlook account used to create the application at `https://portal.azure.com/`. If everything is correct, you are prompted for the Oracle Integration credentials.

- a. Enter the credentials and click **OK**. These are the same credentials you use to log in to Oracle Integration.
- b. Enter the Microsoft Office 365 Outlook account credentials.
A page is displayed asking for permission to interact with the account.
- c. Click **Yes**.
The Access Allowed! page is displayed.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

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

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Integration Flow

Edit the **Oracle PDF Invoice Create** integration flow to configure the document understanding action.

To know more about the document understanding action, see Extract Document Information with a Document Understanding Action in *Using Integrations in Oracle Integration 3*.

1. In the Integrations section, click the integration flow name.
2. On the integration canvas, click the **Document Understanding** element, and then click **Actions** . . . and select **Edit**.
3. Click **Next** and select the OCI compartment.
4. Click **Next** and then **Done**.
5. Click **Save** to save the integration.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **Generic_Properties** lookup table and enter appropriate values.

1. In the Lookups section, click the lookup name.
2. Enter appropriate values against each of the following properties.

PropertyName	PropertyValue
FromAddress	Enter the email address from which notification emails have to be sent in case of errors or failures.
ToAddress	Enter the email address to which notification emails have to be sent in case of errors or failures.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** . . . on the **Oracle Invoice PDF Processing Scheduler** integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the lastRun parameter. This parameter stores the date of the most-recent successful run of the integration flow. For the initial run, it contains a default value. The parameter's value is automatically updated after each successful run, and only the invoices received after the date stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date for a specific scenario, enter the date of your choice in the **New Value** field in the format, yyyy-MM-dd.

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle ERP Cloud instance and check for the new invoices created.

Related Documentation

- *Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*
- [Oracle Cloud Infrastructure Document Understanding](#)

Create Jira Issues for Exceptions in Integrations

Use this recipe to automatically create a Jira issue when an exception is raised in an integration.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates a process for creating issues in Jira, in response to integration failures. Any integration experiencing an exception may invoke this recipe, passing the requisite information to facilitate Jira issue creation.

To use the recipe, you must install the recipe and configure the connection and other resources within it. When an exception occurs in an integration, the recipe sends a request to your Jira instance with the captured exception data. Subsequently, an issue is created in Jira with the details provided in the request payload.

System and Access Requirements

- Oracle Integration 3
- Atlassian
- An account on Atlassian with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Atlassian instance in order to successfully connect to it using Oracle Integration and create Jira issues for exceptions in integrations.

Create an API Token in Your Atlassian Account

You'll require an API token to connect to your Atlassian instance from Oracle Integration.

To create an API token:

1. Log in to your Atlassian account using the following URL: <https://id.atlassian.com>.
2. On the Start page, click **Account Settings**.
3. On the left navigation pane, click **Security**.
4. 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**.
6. 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.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Jira Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter your Atlassian instance URL in the **Host name** field. For example: <https://myinstance.atlassian.net/>.
3. 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 .

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Create Issue in Jira** integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click the **Body** tab and enter the request data.

Example:

```
{
  "summary": "Integration issue reported test-id",
  "issueTypeName": "Task",
  "projectKey": "OP",
  "description": "epic test\n\n*dssdf*",
  "reporterId": "712020:04e44ed7-d804-456f-b566-8b831a3031eb"
}
```

Note

Enter the correct **projectKey** and **issueTypeName** as per your Jira environment.

- c. Click **Run**.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify that a new Jira issue is created with the required information.
 - a. Log in to your Atlassian instance: `https://<hostname>.atlassian.net/`.
 - b. Open the project in which the issue is created.
 - c. Open the newly created Jira issue and check if the details match the information passed in the request payload.

Related Documentation

- [Using the Jira Adapter with Oracle Integration 3](#)

Create Jira Issues for Project Tasks in Oracle NetSuite

Use this recipe to create Jira issues for project tasks in Oracle NetSuite.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)

- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates issues in Jira for project tasks in Oracle NetSuite as per a schedule specified in Oracle Integration. For the project tasks to be picked by the recipe, they must be associated with a company and must not already be mapped to a Jira issue ID.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads all the project tasks meeting the criteria from the Oracle NetSuite instance and, in turn, creates corresponding issues on your Jira instance. Additionally, the recipe updates the Jira issue IDs in the corresponding project tasks in Oracle NetSuite.

Basic data associated with Jira issues, such as issue type, summary, priority, description, status, and comments are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462) or higher
- Oracle NetSuite
- Atlassian
- An account on Oracle NetSuite with the Administrator role
- An account on Atlassian 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
- An Oracle NetSuite project with at least one project task on your Oracle NetSuite account

Before You Install the Recipe

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

Configure Jira

To access Jira from Oracle Integration, you must perform the following configuration tasks:

1. Create an app, configure the OAuth 2.0 authorization code, and assign the scopes. See *Prerequisites to Use the Authorization Code Credentials Security Policy in Using the Jira Adapter with Oracle Integration 3*.
2. Create a custom field in Jira to store the Oracle NetSuite project task ID. See [Create a new custom field](#). Specify the name of the field as NetSuiteProjectTaskId.
3. Obtain the project key of the Jira project in which you want the issues to be created. On your Jira instance:
 - a. Click **Projects**, and then **View All Projects**. A list of all projects you have access to is displayed. Select the project from the list.
 - b. Click **Summary** to view the Activity page. Note the **Key** displayed on the right pane.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration and create project tasks, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy. In addition, create a custom field to store the Jira issue ID.

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

1. Perform the general configurations necessary to connect to Oracle NetSuite. See [Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).
2. Perform the TBA-related configuration tasks. See [Prerequisites for the Token-Based Authentication Security Policy in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Transactions	Adjust Inventory	Full
Transactions	Cash Sale	Full
Transactions	Cash Sale Refund	Full
Transactions	Customer Deposit	Full
Transactions	Customer Payment	Full
Transactions	Customer Refund	Full
Transactions	Fulfill Orders	Full
Transactions	Invoice	Full
Transactions	Item Shipment	Full
Transactions	Sales Order	Full
Transactions	View Payment Events	Full
Reports	Integration	View
Reports	Inventory	View
Reports	Sales	View
Lists	Accounts	Full
Lists	Cases	Full
Lists	Companies	Full
Lists	Contacts	Full
Lists	Custom Record Entries	Full
Lists	Customers	Full
Lists	Employees	Full
Lists	Items	Full
Lists	Perform Search	Full
Setup	REST Web Services	Full
Setup	SOAP Web Services	Full

Tab Name	Permission	Level
Setup	User Access Tokens	Full

3. Obtain the internal ID of the project from which the recipe must read project tasks. From your home page, select **Lists**, then **Relationships**, and then **Projects**. Select the project and note the internal ID.
4. Create a custom field in your Oracle NetSuite instance to store the Jira issue ID. See [Creating Custom CRM Fields](#). While creating a custom field:
 - a. Specify the value for **Label** as JiraTaskId and **ID** as _jirataskid.
 - b. Check the **PROJECT TASK** box in the **Applies To** section.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Jira Connection

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

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

Field	Information to Enter
Security Policy	Select Authorization Code Credentials .
Client ID	Enter the integration key of the OAuth application created in Jira. See Configure Jira .
Client Secret	Enter the client secret of the OAuth application created in Jira.
Scope	Enter read:jira-work, manage:jira-project, write:jira-work, manage:jira-configuration, read:jira-user.

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

You'll now be redirected to the Jira login page.

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

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

8. On the Jira web page, click **ACCEPT**.

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

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

A message confirms if your test is successful.

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

Configure the Oracle NetSuite Connection

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

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

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, https://webservices.netsuite.com/wsd/v2022_1_0/netsuite.wsdl . See Configure Oracle NetSuite .

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

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier.

Note

You must enter this information in capital letters.

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

A message confirms if your test is successful.

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

Update Integration Properties

You have to update the integration property values so that the Jira issues are created in the right project.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, update the integration properties with appropriate values.
 - a. **NetSuiteProjectInternalId**: This integration property holds the internal ID of the Oracle NetSuite project with the project tasks. Enter the internal ID you obtained earlier. See [Configure Oracle NetSuite](#).
 - b. **FromAddress**: This integration property holds the email address from which run-time exception emails are sent. Enter an email address of your choice.
 - c. **ToAddress**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
 - d. **JiraProjectKey**: This integration property holds the key of the Jira project in which the issues are created. Enter the key you obtained earlier. See [Configure Jira](#).
3. 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 other resources, you can activate and run the recipe.

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Jira instance and check for new issues.
 - a. Click **Projects**, and then **View All Projects**. A list of all projects you have access to is displayed. Select the configured project from the list.
 - b. From your project's sidebar, select **Issues and filters**.
 - c. Select **All issues** to view all issues in the project.
5. Log in to your Oracle NetSuite instance and check if the Jira issue IDs are updated in the project tasks. From your home page, select **Lists**, then **Relationships**, and then **Projects**. Select the project and verify the value of the **JiraTaskId** field.

Related Documentation

- [Using the Oracle NetSuite Adapter with Oracle Integration 3](#)

- *Using the Jira Adapter with Oracle Integration 3*

Create Jira Issues for ServiceNow Incidents

Use this recipe to create corresponding Jira issues for ServiceNow incidents.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates a corresponding issue in Jira when a new incident is created in ServiceNow.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When an incident is created in ServiceNow, the integration flow of the recipe is triggered, and it creates a corresponding Jira issue. Basic data is synchronized between the two platforms.

System and Access Requirements

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

Before You Install the Recipe

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

Create an API Token in Your Atlassian Account

To access Jira from Oracle Integration and create corresponding Jira issues for ServiceNow incidents, you'll require an API token associated with your Atlassian account.

To create an API token:

1. Log in to your Atlassian account using the following URL: <https://id.atlassian.com>.
2. On the Start page, click **Account Settings**.
3. On the left navigation pane, click **Security**.

4. 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**.
6. 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 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 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle REST Jira Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter your Atlassian instance URL in the **Host name** field. For example: `https://myinstance.atlassian.net/`.
3. 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 .

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

A message confirms if your test is successful.

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

Configure the Oracle ServiceNow Connection

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

2. In the Properties section, enter your ServiceNow instance URL in the **ServiceNow Instance Name** field. For example, `https://<instance_name>.service-now.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the username of your ServiceNow instance. See Configure ServiceNow .
Password	Enter the password of your ServiceNow instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table to enter the necessary details.

1. In the Lookups section, click the lookup name.
2. Against the **EmailTo** field, enter the email address to which runtime exception emails are to be sent.
3. In the **ServiceNow_Value** column, enter the category of the incident created in ServiceNow.
4. In the **Jira_Value** column, enter the project key of your Jira project.
5. Click **Save**. If prompted, click **Save** again.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Configure the basic authentication in ServiceNow for the integration flow of the recipe.
 - a. After activating the recipe, obtain the metadata URL of the integration flow.
 - i. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run details**.
 - ii. In the Run details panel, copy the **Metadata URL** value.
 - b. Use the metadata URL to configure a basic authentication in ServiceNow. This is necessary to invoke the integration flow in Oracle Integration from ServiceNow.
See *Invoke an Integration from ServiceNow* in *Using the ServiceNow Adapter with Oracle Integration 3*.
3. Run the recipe.

To run the recipe, you must perform the following tasks in the ServiceNow instance.

- a. Log in to your ServiceNow instance as an 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**.

You've now created an incident in ServiceNow and triggered the recipe.
4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Check for the new Jira issue created.
 - a. Log in to your Atlassian instance.
 - b. Select **Search** in the navigation bar.
 - c. Choose the newly created issue from the list of recent issues that are displayed. Open the issue and check if the details match the incident that you created in ServiceNow.
 - d. Optionally, you can filter the search results by choosing the project under which the issue is created.

Related Documentation

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

Create New User Records in Azure Active Directory

Use this recipe to create new user records in Azure Active Directory.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates new user records in Azure Active Directory, based on the data received from an external system through REST APIs.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually. To run the integration flow, you must provide the user record details through a POST request via a REST API call from an external application. Subsequently, the integration flow of the recipe is triggered, and it creates a new user record in Azure Active Directory based on the request payload. If a similar user record already exists in Azure Active Directory, then it skips user record creation and a response message is returned along with status-203.

System and Access Requirements

- Oracle Integration 3
- Azure Active Directory
- An account on Azure Active Directory with the Administrator role

Before You Install the Recipe

To access Azure Active Directory using Oracle Integration and create new user records, you must perform certain configuration tasks on your Azure Active Directory instance.

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

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Azure Active Directory Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the tenant ID that you obtained in the **Tenant ID** field. See Prerequisites for Creating a Connection in *Using the Azure Active Directory Adapter with Oracle Integration 3*.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Client Credentials .
Client ID	Enter the client ID. See Prerequisites for Creating a Connection in <i>Using the Azure Active Directory Adapter with Oracle Integration 3</i> .
Client Secret	Enter the client secret.

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

A message confirms if your test is successful.

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

Configure the Lookup Table

The recipe contains the following lookup tables that you can edit as necessary.

- **ORACLE-BRT_AZURE_AD_LOOKUP**: Use this lookup table to enable user account, configure the domain associated with the user principal name (UPN) used for identification, and to set password sign in.
- **ORACLE-BRT-EMAIL_CONSTANTS**: Use this lookup table to configure email properties for error handling.

1. In the Lookups section, click the lookup name.
2. Configure the lookup table.
 - a. In the **ORACLE-BRT_AZURE_AD_LOOKUP** lookup table, edit the following properties as necessary.

Property	Description
AccountEnabled	Specify the value as either True or False.
UserPrincipalNameDomain	Specify the user name domain. For example: john.doe@example.com
PasswordSignIn	Specify the value as either True or False.

- b. In the **ORACLE-BRT-EMAIL_CONSTANTS** lookup table, edit the following properties as necessary.

Property	Description
From	Specify the email address from which error notifications have to be sent.
To	Specify the email address to which error notifications have to be sent.
Subject	Specify the subject of the error notification email. For example: Oracle Azure User Sync failed

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the request payload.

Example:

```
{
  "Department": "",
  "Email": "",
  "Address": {
    "State": "",
    "Street": "",
    "Country": "",
    "City": ""
  },
  "BusinessTitle": "",
  "FirstName": "",
  "LastName": "",
  "EmployeeID": ""
}
```

Based on the data in the request payload, the recipe creates a new record in Azure Active Directory and provides the required response.

If a similar user record already exists in Azure Active Directory, a response message User already exists- <Existing User Principal Name> along with status-203 is returned.

If a new user record is successfully created in Azure Active Directory, a response message User Created-<User Principal Name> along with status-201 is returned.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

Related Documentation

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

Create Oracle CPQ Transactions for Oracle Intelligent Advisor Interviews

Use this recipe to create transactions in Oracle CPQ for Oracle Intelligent Advisor interviews.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates a corresponding transaction in Oracle CPQ from information based on an Oracle Intelligent Advisor interview.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a customer launches an interview form in Oracle Intelligent Advisor, the first integration flow gets triggered in Oracle Integration. The integration then loads product catalog (items and quantity) from Oracle CPQ in the interview form. In Oracle Intelligent Advisor, the customer completes the form by entering personal details (such as first name, last name, email), choosing the required items and quantity (that are loaded from Oracle CPQ) and

then submitting the form. Subsequently, the second integration flow gets triggered and it creates a corresponding new transaction in Oracle CPQ based on the details submitted by the customer in the Oracle Intelligent Advisor interview form. An email with quotation based on the items and quantities selected in the interview form is also sent to the customer.

System and Access Requirements

- Oracle Integration, Version 24.08 or higher
- Oracle Intelligent Advisor
- An account on Oracle Intelligent Advisor with the Administrator role
- Oracle CPQ
- An account on Oracle CPQ with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Intelligent Advisor and Oracle CPQ instances in order to successfully connect to these external systems using Oracle Integration and create Oracle CPQ transactions for Oracle Intelligent Advisor interviews.

Configure Oracle Intelligent Advisor

To access Oracle Intelligent Advisor using Oracle Integration and create Oracle CPQ transactions for Oracle Intelligent Advisor interviews, you must perform certain configurations on your Oracle Intelligent Advisor instance.

1. Import the project into Policy Modeling, the interview designer.
Download the sample project file to import - [Customer Product Quote.zip](#).
2. Deploy the project into Oracle Intelligent Advisor Hub.
3. [Create a connection to Oracle Integration from Intelligent Advisor Hub](#) using OAuth.
4. [Create a confidential application to connect to Oracle Integration](#).

For more information, see [Use the Intelligent Advisor Adapter for Interviews](#).

Configure Oracle CPQ

To access Oracle CPQ using Oracle Integration, you must perform certain configurations on your Oracle CPQ instance.

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

1. Create a user account for Oracle Integration.
2. Obtain the REST Catalog URL. See Prerequisites for Creating a Connection in *Using the Oracle CPQ Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Intelligent Advisor Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the Oracle Intelligent Advisor base URL in the **Intelligent Advisor Hub URL** field. For example: `https://{your_site_interface}/opa-hub`
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Enter <code>https://{your_site_interface}/opa-hub/api/{version}/auth</code>
Client Id	The name of the API client in Intelligent Advisor Hub.
Client Secret	The password for the API client in Intelligent Advisor Hub.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle CPQ Connection

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

Field	Information to Enter
Connection Type	Select REST Catalog URL .
Connection URL	Enter the base URL of your Oracle CPQ instance.

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

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on the Oracle CPQ instance.
Password	Enter the password of the account created for Oracle Integration on the Oracle CPQ instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

The recipe contains the following lookup tables: **ORCL-BRT-CPQ_STATE_LOOKUP** and **ORCL-BRT-CPQ_COUNTRY_LOOKUP**.

1. In the Lookups section, click the lookup name.
2. Edit the lookup tables with appropriate values.
 - The **ORCL-BRT-CPQ_STATE_LOOKUP** lookup table contains state code and state name. If you have Customers which have addresses that contain different states than the ones present in the lookup, add them in the lookup table.
 - The **ORCL-BRT-CPQ_COUNTRY_LOOKUP** lookup table key-value pairs of country code and country description. If you have Customers which have addresses that contain different countries than the ones present in the lookup, add them in the lookup table.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties of the integration flows in the recipe.

1. In the Integrations section, click **Actions . . .** on the **Oracle IA CPQ Save CPQ Transaction** integration flow, then select **Update property values**.
2. In the Update property values panel, update the **pricebookvarname** integration property by providing a valid price book name in the **New value** field.
3. Click **Submit**.

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

4. In the Integrations section, click **Actions . . .** on the **Oracle IA CPQ Load CPQ Transaction** integration flow, then select **Update property values**.
5. In the Update property values panel, update the following integration properties.
 - **pricebookvarname**: This integration property holds the price book name.
 - **limit**: This integration property holds the requested page size. The property sets the limit for the maximum number of elements that the collection should return.
 - **offset**: This integration property holds the offset of the page. By default, the offset value is zero which means that the first page will be returned.
 - **q**: This integration property holds the product/item name. For example: AS09743 - Vision Cloud 250gb Backup.
6. Click **Submit**.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).

2. Run the recipe.
 - a. Log in to your Oracle Intelligent Advisor instance and launch an interview.

This triggers the **Oracle IA CPQ Load CPQ Transaction** integration flow which loads the product catalog (products and quantity) from Oracle CPQ into the interview's Product page.
 - b. Enter customer information such as first name, last name, email, phone, address on the Customer page, and click **Next**.

To test the recipe, you can enter your email in the customer email address field. This will ensure that you get an email with the quotation for the products selected, on successful run of the recipe.
 - c. On the Sales person page, the sales person details appear. Click **Next**.
 - d. On the Product page, choose the products and corresponding quantity for each product.
 - e. Click **Submit**.

On submitting the interview in Oracle Intelligent Advisor, the **Oracle IA CPQ Save CPQ Transaction** integration flow is triggered. On successful execution of the integration flow, a new transaction corresponding to the submitted Oracle Intelligent Advisor interview is created in Oracle CPQ.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to the Oracle CPQ instance and verify that a new transaction corresponding to the Oracle Intelligent Advisor interview is created.

Also, check if you received an email with the quotation for the new transaction.

Related Documentation

- *Using the Oracle Intelligent Advisor Adapter with Oracle Integration 3*
- *Using the Oracle CPQ Adapter with Oracle Integration 3*

Create Orders in Oracle Fusion Cloud Service for Oracle CPQ Opportunities

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

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 Fusion Cloud Service.

Transaction data is passed seamlessly between the two applications, ensuring accuracy and allowing Oracle Fusion Cloud 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.

Note

This is a package-based recipe. See [About Packages and Edit and Replace Dependent Resources in a Package](#)

System and Access Requirements

- Oracle Fusion Cloud Service, Release 9 or later
- Oracle CPQ, 2015 R1 or later
- Oracle Integration

Install, Configure, and Run the Recipe

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

Configure resources of the recipe. See [Configure Resources in a Package](#).

Related Documentation

[Oracle CPQ Cloud-Oracle Sales Cloud Integration through Oracle Integration Cloud Service Implementation Guide](#)

Create Products in Magento Commerce Cloud and QuickBooks for Oracle ERP Cloud Items

Use this recipe to create products in Magento Commerce Cloud and QuickBooks for Oracle ERP Cloud items through GCP Pub Sub.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates a product in Magento Commerce Cloud and QuickBooks for an item created in Oracle ERP Cloud through GCP Pub Sub.

To use the recipe, you must install the recipe and configure the connections and other resources within it. There are three application-driven integration flows in the recipe. The first integration flow is triggered when a new item is created in Oracle ERP Cloud. It fetches the new item and publishes it to GCP Pub Sub. This triggers the second and third integration flows. Subsequently, the second and third integration flow creates a corresponding product in Magento Commerce Cloud and QuickBooks respectively for the Oracle ERP Cloud item.

System and Access Requirements

- Oracle Integration, Version 24.08 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- GCP Pub Sub
- An account on GCP Pub Sub with the Administrator role
- Magento Commerce Cloud
- An Integration user on Magento Commerce Cloud with access to all resources
- QuickBooks
- An account on QuickBooks with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud, GCP Pub Sub, Magento Commerce Cloud and QuickBooks instances in order to successfully connect to these external systems using Oracle Integration and create products for items.

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.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure GCP Pub Sub

To successfully connect to GCP Pub Sub using Oracle Integration and publish products, you must perform certain general configurations on your GCP Pub Sub instance. See *Prerequisites for Creating a Connection in Using the GCP Pub Sub Adapter with Oracle Integration 3*.

Also, you must create a topic to store the Oracle ERP Cloud items that are published. See [Create a topic](#).

Configure Magento Commerce Cloud

To access the Magento Commerce Cloud instance from Oracle Integration and create products, you will need to create a user account for Oracle Integration on Magento Commerce Cloud. See *Prerequisites for Creating a Connection in Using the Magento Commerce Cloud Adapter with Oracle Integration 3*.

Configure QuickBooks

To access the QuickBooks instance from Oracle Integration and create products, you'll need to create a user account for Oracle Integration on QuickBooks. See *Prerequisites for Creating a Connection in Using the QuickBooks Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ERP Cloud Connection

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

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

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

A message confirms if your test is successful.

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

Configure the GCP Pub Sub Invoke Connection

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

Field	Information to Enter
Project Id	Enter your GCP Pub Sub project Id. See Configure GCP Pub Sub .
Service Account	Enter GCP Pub Sub service account detail.

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

Field	Information to Enter
Google Client ID	Enter the Google client ID. See Configure GCP Pub Sub .
Google Client Secret	Enter the Google client secret.

- Click **Provide Consent** to verify the connection properties and get an access token.
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

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

Configure the GCP Pub Sub Trigger Connection

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

Field	Information to Enter
Project Id	Enter your GCP Pub Sub project Id. See Configure GCP Pub Sub .
Service Account	Enter GCP Pub Sub service account detail.

- In the Security section, enter the **JWT Private Key Alias**. See [Configure GCP Pub Sub](#).
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

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

Configure the Magento Commerce Cloud Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the Magento Commerce Cloud host name.
- In the Security section, enter the following details:

Field	Information to Enter
Consumer Key	Enter the consumer key value. See Configure Magento Commerce Cloud .
Consumer Secret	Enter the consumer secret value.
Access Token	Enter the access token value.
Access Token Secret	Enter the access token secret value.

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

A message confirms if your test is successful.

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

Configure the Lookup Tables

Edit the lookup tables of the recipe with appropriate values.

1. In the Lookups section, click the lookup name.
2. The recipe contains the following two lookup tables.
 - a. **MagentoLookup**: Use this lookup to configure Magento request parameters.

Property	Description	Value
Media Type	Specify the media type of the attachment.	image
Label	Specify the value of the label.	image
Disabled	Specify the value either True or False.	false
Types1	Specify the type of the attachment.	image
Types2	Specify the type of the attachment.	image_small
Types3	Specify the type of the attachment.	thumbnail

- b. **QuickBooksLookup**: Use this lookup to configure the QuickBooks request parameters.

Property	Description	Value
Income Account Ref Val	Specify the income account reference ID.	79
Income Account Ref Name	Specify the income account reference name.	Sales of Product Income
Expense Account Ref Val	Specify the expense account reference ID.	80
Expense Account Ref Name	Specify the expense account reference name.	Cost of Goods Sold
Asset Account Ref Val	Specify the asset account reference ID.	81

3. In the **Value** column of the lookup tables, enter or map the appropriate values for the properties.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).

2. Update the integration properties for each of the three integration flows of the recipe: **Oracle ERP Cloud GCP Pub/Sub Product Sync**, **GCP Pub/Sub QuickBooks Product Sync** and **GCP Pub/Sub Magento Product Sync**.
 - a. In the Integrations section, click **Actions** . . . on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo**: This integration property holds the email address to which error notifications are to be sent. Enter an email of your choice.
 - **retryCount**: This integration property holds the number of retries when an invoke fails. The default value is 3.
 - c. Click **Submit**.
3. Run the recipe.

To run the recipe, you must create an item in Product Information Management under Product Management in Oracle ERP Cloud.

 - a. Click **Create Item**, under tasks in the Product Information Management section.
 - b. Enter the organization.
 - c. Enter the number of items.
 - d. Select the item class.
 - e. Select the template from the available list.
 - f. Click **OK**.
 - g. Enter the item name on the new page that appears. Enter the item's description.
 - h. Enter other mandatory details like Primary Unit of Measure.
 - i. Click **Save**.

You've now created an item in Oracle ERP Cloud and successfully triggered the first integration flow of the recipe. Subsequently, the second and third integration flows of the recipe are triggered when the newly created item is published to GCP Pub Sub.
4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to the Magento Commerce Cloud instance, and check for the new product created.
 - a. Navigate to **Catalogue**. Under Catalogues, select **Product** and enter the product name.
 - b. Fetch the product by order id or order number.
6. Log in to the QuickBooks instance, and check for the new product created.
 - a. Navigate to the Intuit Developer section.
 - b. In the Sandbox, search for the term `items`, and get the list of items created.

Troubleshooting Tip: If the service target is not reachable due to token expiration, retest the connection.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the GCP Pub Sub Adapter with Oracle Integration 3*
- *Using the Magento Commerce Cloud Adapter with Oracle Integration 3*

- *Using the QuickBooks Adapter with Oracle Integration 3*

Create Products in Shopify for Items in Oracle ERP Cloud

Use this recipe to create corresponding products in Shopify when items are created in Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a product in Shopify whenever an item is created in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When an item record is created in the Oracle ERP Cloud instance, the recipe is triggered, and it creates a corresponding product record in Shopify. Basic data, such as attachments, price, title, and product type are synchronized between the two platforms.

System and Access Requirements

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

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Shopify instances in order to successfully connect to these external systems using Oracle Integration and transfer product records between them.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.

2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Shopify

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

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ERP Cloud Connection

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

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

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

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

Configure the Shopify Connection

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

Field	Information to Enter
Host Name	Enter the Shopify host name. The Shopify host name appears in the Shopify home URL. For example: <code>https://hostname.myshopify.com</code> .
Shopify REST API Version	Enter the REST API version.

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

Field	Information to Enter
Security Policy	Leave Shopify Security Policy selected.
Username	Enter the API key obtained earlier. See Configure Shopify .
Password	Enter the password obtained earlier.
Shared secret	Enter the shared secret obtained earlier.

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must perform the following tasks in the Oracle ERP Cloud instance.

- a. On the home page, click **Menu** .
- b. From the list, select **Product Management**, then click **Product Information Management**.
- c. On the Product Information Management page, click **Tasks**, then **Create Item**.
- d. On the Create Item page, select the required organization and item templates, and click **OK**.
- e. On the Item Details page, enter all the mandatory information for the item, like name, description, and so on.
 - i. On the **Overview** tab, enter the unit of measure.
 - ii. On the **Specifications** tab, enter the pricing details.
 - iii. On the **Attachments** tab, add an attachment to the item record.
 - iv. Click **Save**.

You've now created an item record in Oracle ERP Cloud and triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Shopify instance and check for the new product created.
 - a. On the home page, select your Shopify store. For example, `orcl-shop-connector-dev-store`.

- b. On the store page, click Products on the left navigation pane.

The list of products is displayed, which contains the corresponding product record for the item created in Oracle ERP Cloud.

Related Documentation

- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Create Project Tasks in Oracle NetSuite for Jira Issues

Use this recipe to create project tasks in Oracle NetSuite for task issues in Jira.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a project task in Oracle NetSuite whenever a task issue is created in Jira.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a new task issue is created in Jira, the integration flow of the recipe is triggered, and it creates a corresponding project task in the Oracle NetSuite instance. Basic data is synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle NetSuite
- Atlassian
- An account on Oracle NetSuite with the Administrator role
- An account on Atlassian 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 Oracle NetSuite project on your Oracle NetSuite account

Before You Install the Recipe

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

Configure Jira

To access Jira from Oracle Integration, you must perform the following configuration tasks:

1. Create an app, configure the OAuth 2.0 authorization code, and assign the scopes. See *Prerequisites to Use the Authorization Code Credentials Security Policy in Using the Jira Adapter with Oracle Integration 3*.
2. Create a custom field in Jira to store the Oracle NetSuite project task ID. See [Create a new custom field](#). Specify the name of the field as NetSuiteProjectTaskId.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration and create project tasks, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

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

1. Perform the general configurations necessary to connect to Oracle NetSuite. See *Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3*.
2. Perform the TBA-related configuration tasks. See *Prerequisites for the Token-Based Authentication Security Policy in Using the Oracle NetSuite Adapter with Oracle Integration 3*.

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Transactions	Adjust Inventory	Full
Transactions	Cash Sale	Full
Transactions	Cash Sale Refund	Full
Transactions	Customer Deposit	Full
Transactions	Customer Payment	Full
Transactions	Customer Refund	Full
Transactions	Fulfill Orders	Full
Transactions	Invoice	Full
Transactions	Item Shipment	Full
Transactions	Sales Order	Full
Transactions	View Payment Events	Full
Reports	Integration	View
Reports	Inventory	View
Reports	Sales	View
Lists	Accounts	Full
Lists	Cases	Full
Lists	Companies	Full
Lists	Contacts	Full
Lists	Custom Record Entries	Full
Lists	Customers	Full
Lists	Employees	Full
Lists	Items	Full

Tab Name	Permission	Level
Lists	Perform Search	Full
Setup	REST Web Services	Full
Setup	SOAP Web Services	Full
Setup	User Access Tokens	Full

3. Create a project in your Oracle NetSuite instance. See [Creating a Project Record](#).
4. Create a custom field in your Oracle NetSuite instance. See [Creating Custom CRM Fields](#). While creating a custom field:
 - a. Specify the value for **Label** as JiraTaskId and **ID** as _jirataskid.
 - b. Check the **PROJECT TASK** box in the Applies To section.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Jira Connection

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

Field	Information to Enter
Security Policy	Select Authorization Code Credentials .
Client ID	Enter the integration key of the OAuth application created in Jira. See Configure Jira .
Client Secret	Enter the client secret of the OAuth application created in Jira.
Scope	Enter read:jira-work, manage:jira-project, write:jira-work, manage:jira-configuration, read:jira-user.

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

You'll now be redirected to the Jira login page.

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

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

8. On the Jira web page, click **ACCEPT**.

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

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

A message confirms if your test is successful.

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

Configure the Oracle NetSuite Connection

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

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, <code>https://webservices.netsuite.com/wSDL/v2022_1_0/netsuite.wsdl</code> , where <code><NetSuite_application_version></code> is the version of the Oracle NetSuite application. For example, <code>v2015_1_0</code> . See Configure Oracle NetSuite .

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

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier.

Note

You must enter this information in capital letters.

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

- a. Create an issue in Jira. See [Create an issue](#).
- b. While creating the issue, choose the issue type as Task.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check for the new project task created in Oracle NetSuite. See [Task Views](#).

Related Documentation

- *Using the Oracle NetSuite Adapter with Oracle Integration 3*
- *Using the Jira Adapter with Oracle Integration 3*

Create Sales Orders in Oracle ERP Cloud for Salesforce Opportunities

Use this recipe to create sales orders in Oracle ERP Cloud for Salesforce opportunities won.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates sales orders in Oracle ERP Cloud whenever an opportunity record closes with the status *Closed Won* in Salesforce.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a Salesforce opportunity record closes with the status *Closed Won*, the integration flow of the recipe is triggered. Initially, it checks if a corresponding customer record (for the Salesforce account) exists in Oracle ERP Cloud. If not, a new customer record is created. Subsequently, the flow creates a corresponding sales order in the Oracle ERP Cloud instance for that customer.

Note

The products catalog must be synchronized in advance between Oracle ERP Cloud and Salesforce. In addition, the opportunity in Salesforce must contain a product as a line item.

System and Access Requirements

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

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration and create sales orders for opportunities won.

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve information about the status of Salesforce opportunities, you must perform certain configurations 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. Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce opportunity is closed with the status Closed Won in Salesforce. Finally, create a custom field for Salesforce opportunity and product records to store Oracle ERP Cloud IDs.

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

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
 - a. On the Salesforce Setup page:
 - i. Expand **Manage Users** under the Administer section in the left navigation pane.
 - ii. Click **Profiles**.
 - b. On the Profiles page, click **New Profile**.
 - c. On the resulting page:
 - i. Select **Standard User** in the **Existing Profile** field.
 - ii. Enter a name for the new profile, for example, API Enabled, and click **Save**.
The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.

- In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
- iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

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

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
- i. Enter a first name and last name for the user, for example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce.
You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.
Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
- i. Stay signed in as the new user and click the user name at the top of the page to open a menu.

- ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account.
Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
5. Create an outbound message. The outbound message triggers the integration flow of the recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**. If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.
 - c. On the New Outbound Message page:
 - i. In the **Object** field, select **Opportunity** and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter a name for the outbound message.
Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the outbound 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_ERP_OPPOR_TO_ORDER/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**, 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 opportunity record closes with the status *Closed Won*.
 - 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 **Opportunity** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:
 - Enter the rule's name.
 - Optionally, enter a description for the rule.
 - In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
 - In the Rule Criteria section, select the **formula evaluates to true** option, and define the following rule criteria:
Opportunity: Stage EQUALS Closed Won
 - Click **Save & Next**.
 - iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
 - iv. On the Select Existing Actions page:
 - Select **Outbound Message** in the **Search** field.
 - In the **Available Actions** box, select the outbound message you created previously, for example, Outbound Message: <name of your outbound message>.
 - Click **Add** to move your selection to the **Selected Actions** box.
 - Click **Save**.
 - v. Click **Done**.
The new workflow rule is created and displayed on your browser.
 - d. Click **Activate** to activate the rule.
7. Create a custom field for opportunity records.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Opportunities**, and then click **Fields**.
 - b. On the Opportunity Fields page, scroll to the Opportunity Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter ERPCustomerId as the field label. Note that the field name is automatically populated based on the label you enter.
 - Enter 255 as the length.
 - Optionally, enter a description for the new field.

- 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**.
- d. Create another custom field for Opportunity records by repeating Steps b and c, but enter a different label name, for example ERPOrderId.

On the Opportunity Fields page, you can see the new fields added under the Opportunity Custom Fields & Relationships section.

8. Create a custom field for product records.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Products**, and then click **Fields**.
 - b. On the Product Fields page, scroll to the Product Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter ERPID as the field label. Note that the field name is automatically populated based on the label you enter.
 - Enter 255 as the length.
 - Optionally, enter a description for the new field.
 - 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 Product Fields page, you can see the new field added under the Product Custom Fields & Relationships section.

Configure Oracle ERP Cloud

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

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Salesforce Connection

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

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

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

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce.

Note

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

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

A message confirms if your test is successful.

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

Configure the Oracle ERP Cloud Connection

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

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the lookup table and map the country codes between Salesforce and Oracle ERP Cloud instances.

1. In the Lookups section, click the lookup name.
2. Map the countries in the Salesforce instance to the corresponding country codes in the Oracle ERP Cloud instance.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page (in the Classic UI), click the **Opportunities** tab.
 - c. On the Opportunities Home page, select **All Opportunities** in the **View** field and click **Go!**.
All the opportunities in your Salesforce instance are displayed.
 - d. Find a sample opportunity record that hasn't been closed, and click its name to open it.
 - e. On the opportunity's page, in the Opportunity Detail section at the top, click **Edit**, then select **Closed Won** in the **Stage** field, and click **Save**.

You have now successfully triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle ERP Cloud instance as an Administrator, and check for the new sales order record.
 - a. On the Oracle ERP Cloud home page, select the **Order Management** tab, and then click **Order Management**.

- b. On the resulting page, use the Search (or Advanced) field to find the required sales order.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Create Sales Orders in Oracle NetSuite for Invoices in Magento Commerce Cloud

Use this recipe to create sales orders in Oracle NetSuite for invoices in Magento Commerce Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates sales orders in Oracle NetSuite for Magento Commerce Cloud invoices, as per a schedule specified in Oracle Integration.

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

When triggered, the integration flow reads all the invoices from the configured store in the Magento Commerce Cloud instance and, in turn, creates corresponding sales orders in your Oracle NetSuite instance.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462), or higher
- Magento Commerce Cloud
- An account on Magento Commerce Cloud with the Administrator role
- Oracle NetSuite
- An account on Oracle NetSuite with the Administrator role

Note

Ensure that items are synchronized between the Magento Commerce Cloud and Oracle NetSuite instances.

Before You Install the Recipe

You must perform the following configuration tasks on your Magento Commerce Cloud and Oracle NetSuite instances in order to successfully connect to these external systems using Oracle Integration and create sales orders for invoices.

Configure Magento Commerce Cloud

To access the Magento Commerce Cloud instance from Oracle Integration and create sales orders, you'll need to create a user account for Oracle Integration on Magento Commerce Cloud and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Magento Commerce Cloud Adapter with Oracle Integration 3*.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

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

1. Perform the general configurations necessary to connect to Oracle NetSuite. See *Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3*.
2. Perform the TBA-related configuration tasks. See *Prerequisites for the Token-Based Authentication Security Policy in Using the Oracle NetSuite Adapter with Oracle Integration 3*.

Note

On the Role page, set the name of the role as `OIC NS Integration Role` and set the ID of the role as `_oracle_oic_ns_integ_role`.

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Transactions	Invoice	Full
List	Currency	Full
List	Customers	Full
List	Subsidiaries	Full
List	Contacts	Full
List	Items	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Create a custom field for invoice in your Oracle NetSuite instance. See [Creating Custom Transaction Body Fields](#). While creating a custom field, enter the following values.

- a. Specify the value for **Label** as MagentoStoreID.
- b. Specify the value for **ID** as magento_storeid.
- c. Check the **SALE** box in the Applies To section.
- d. Check the **Main** box in the Display section.
- e. From the **Save** drop-down list, select **Save & Apply to Forms**.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Magento Commerce Cloud Connection

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

Field	Information to Enter
Hostname	Specify the instance URL of your Magento Commerce Cloud instance.

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

Field	Information to Enter
Consumer Key	Enter the consumer key obtained when you configured Magento Commerce Cloud. See Configure Magento Commerce Cloud .
Consumer Secret	Enter the consumer secret obtained when you configured Magento Commerce Cloud.
Access Token	Enter the access token obtained when you configured Magento Commerce Cloud.
Access Token Secret	Enter the access token secret obtained when you configured Magento Commerce Cloud.

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

A message confirms if your test is successful.

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

Configure the Oracle NetSuite Connection

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

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, https://webservices.netsuite.com/wsd/v2023_1_0/netsuite.wsdl .

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

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See Configure Oracle NetSuite .
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier. See Configure Oracle NetSuite .

Note

You must enter this information in capital letters.

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Property	Description
Magento_StoreID	Specify the unique ID of the store in Magento Commerce Cloud.
Netsuite_Customer_Subsiary	Specify the subsidiary of the customer in Oracle NetSuite.

- c. Click **Submit**.

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

3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Oracle NetSuite instance and search for new sales orders.
 - a. From the homepage, navigate to **Transactions** and select **Sales**.
 - b. Select **Enter Sales Orders** and choose **Search**.

Related Documentation

- *Using the Oracle NetSuite Adapter with Oracle Integration 3*
- *Using the Magento Commerce Cloud Adapter with Oracle Integration 3*

Create Sales Orders in Oracle NetSuite for Salesforce Opportunities

Use this recipe to create sales orders in Oracle NetSuite for Salesforce opportunities won.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates an sales orders in Oracle NetSuite whenever an opportunity record closes with the status *Closed Won* in Salesforce.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a Salesforce opportunity record closes with the status *Closed Won*, the integration flow of the recipe is triggered. Initially, it checks if a corresponding customer record (for the Salesforce account) exists in Oracle NetSuite. If not, a new customer record is created. Subsequently, the flow creates a corresponding sales order in the Oracle NetSuite instance for that customer.

Note

The products catalog must be synchronized in advance between Oracle NetSuite and Salesforce. In Salesforce, the opportunity in Salesforce must contain a product as a line item.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Oracle NetSuite, Version 2018.1.0
- An account on Oracle NetSuite with the Administrator role

Assumptions

- You have synchronized the products catalog between Salesforce and Oracle NetSuite, so that internal IDs of NetSuite products are updated in corresponding product records of Salesforce (in a custom field named NetsuiteID).

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Oracle NetSuite instances in order to successfully connect to these external systems using Oracle Integration and create sales orders for opportunities won.

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve information about the status of Salesforce opportunities, you must perform certain configurations 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.

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

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
 - a. On the Salesforce Setup page:
 - i. Expand **Manage Users** under the Administer section in the left navigation pane.
 - ii. Click **Profiles**.
 - b. On the Profiles page, click **New Profile**.
 - c. On the resulting page:

- i. Select **Standard User** in the **Existing Profile** field.
 - ii. Enter a name for the new profile, for example, **API Enabled**, and click **Save**.
The new profile is now saved, and the Profile Detail page of the new profile is displayed.
- d. Click **Edit** on the Profile Detail page.
- e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

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

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user, for example, **Integration User05**.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce.

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

- e. Set a password and note the same.

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

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account. Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
5. Configure the Salesforce platform event. The platform event notifies external systems about changes to the Salesforce opportunity records.
 - 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 Opportunity_PE.
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.

Field	Information to Enter
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.

i. 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 AccountID_PE.
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.

iv. Click **Save**.

g. Create another custom field by repeating Step f, but enter the label name as OpportunityID_PE.

6. 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, NetSuite.
API Name	This field is automatically populated.
Description	If required, enter a 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.

c. Configure the process trigger.

- 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 Opportunity .
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. Define a condition for the criteria.

In the Set Conditions table, configure the following details:

Field	Information to Enter
Field	Search for <code>Opportunity.StageName</code> and select it.
Operator	Select Equals .
Type	Select Picklist .
Value	Select Closed Won .

iv. From the **Conditions** options list, select **All of the conditions are met (AND)**.v. 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**.vi. Click **Save**.

e. Add an 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, provide the following information.
 - From the **Action Type** drop-down list, select **Create a Record**.
 - Enter a name for the action.
 - In the **Record Type** field, enter the name of the platform event created in Step 5.
 - In the Set Field Values section, create two rows with the following values.
Row 1:

Field	Information to Enter
Field	Type the name of the custom field created in sub-step f of Step 5, that is, AccountID_PE.
Type	Select Field Reference .
Value	Select the Account ID of the Opportunity object.

Row 2:

Field	Information to Enter
Field	Type the name of the custom field created in sub-step g of Step 5, that is, OpportunityID_PE.
Type	Select Field Reference .
Value	Select the ID of the Opportunity object.

- iii. 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**.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration and create sales orders, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

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

1. Perform the general configurations necessary to connect to Oracle NetSuite. See *Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3*.
2. Perform the TBA-related configuration tasks. See *Prerequisites for the Token-Based Authentication Security Policy in Using the Oracle NetSuite Adapter with Oracle Integration 3*.

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Transactions	Adjust Inventory	Full
Transactions	Cash Sale	Full
Transactions	Cash Sale Refund	Full
Transactions	Customer Deposit	Full
Transactions	Customer Payment	Full
Transactions	Customer Refund	Full
Transactions	Fulfill Orders	Full
Transactions	Invoice	Full
Transactions	Item Shipment	Full
Transactions	Sales Order	Full
Transactions	View Payment Events	Full
Reports	Integration	View

Tab Name	Permission	Level
Reports	Inventory	View
Reports	Sales	View
Lists	Accounts	Full
Lists	Cases	Full
Lists	Companies	Full
Lists	Contacts	Full
Lists	Custom Record Entries	Full
Lists	Customers	Full
Lists	Employees	Full
Lists	Items	Full
Lists	Perform Search	Full
Setup	REST Web Services	Full
Setup	SOAP Web Services	Full
Setup	User Access Tokens	Full

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Salesforce Connection

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

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

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

Field	Information to Enter
Security Policy	Select Salesforce Username Password Policy .
Username	Enter the username 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** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle NetSuite Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the Oracle NetSuite WSDL URL in the **WSDL URL** field.
For example: `https://webservices.netsuite.com/wsd1/<NetSuite_application_version>/netsuite.wsdl`. Where `<NetSuite_application_version>` is the version of the Oracle NetSuite application.
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier.

Note

You must enter this information in capital letters.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **subsidiaryId_property**: This integration property holds the internal ID of the NetSuite subsidiary to which the new sales orders will belong. Enter the internal ID of the preferred subsidiary.
 - b. **emailFrom_property**: This integration property holds the email address from which run-time exception emails are sent. Enter an email address of your choice.
 - c. **emailTo_property**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
 - d. **emailSubject_property**: This integration property holds the subject of the run-time exception email. Enter a subject line of your choice.
3. Click **Submit**.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page (in the Classic UI), click the **Opportunities** tab.
 - c. On the Opportunities Home page, select **All Opportunities** in the **View** field and click **Go!**.

All the opportunities in your Salesforce instance are displayed.
 - d. Find a sample opportunity record that hasn't been closed, and click its name to open it.

Ensure that the opportunity you select contains a product that is synchronized with Oracle NetSuite. See [Assumptions](#).
 - e. In the Opportunity Detail section of the Opportunity page, click **Edit**, then select **Closed Won** in the **Stage** field, and click **Save**.

You've now successfully triggered the integration flow of the recipe.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle NetSuite instance and check for the new sales order record and customer record (if created).
 - a. To view the sales order: On the Oracle NetSuite home page, select **Transactions**, then **Sales**, then **Enter Sales Orders**, and then **List**.
 - b. To view the customer record: On the Oracle NetSuite home page, select **Lists**, then **Relationships**, and then **Customers**.

Related Documentation

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

Create Sales Receipts in QuickBooks for Oracle NetSuite Invoices

Use this recipe to create sales receipts in QuickBooks for Oracle NetSuite invoices.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to use Oracle Integration to create sales receipts in QuickBooks for invoices in Oracle NetSuite.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the recipe reads the Oracle NetSuite invoices and, in turn, creates corresponding sales receipts in QuickBooks.

System and Access Requirements

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

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite and QuickBooks instances in order to successfully connect to these external systems using Oracle Integration and create sales receipts for invoices.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

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

1. Perform the general configurations necessary to connect to Oracle NetSuite. See [Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3](#).
2. Perform the TBA-related configuration tasks. See [Prerequisites for the Token-Based Authentication Security Policy in Using the Oracle NetSuite Adapter with Oracle Integration 3](#).

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
List	Inventory	Full
List	Customers	Full
List	Subsidiaries	Full
List	Items	Full
List	Invoice	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Create a custom field for invoice in your Oracle NetSuite instance. See [Creating Custom Transaction Body Fields](#).
 - Specify the value for **Label** as QuickBookSyncStatus.
 - Check the **SALE** box in the Applies To section.
 - Check the **Main** box in the Display section.
 - From the **Save** drop-down list, select **Save & Apply to Forms**.
4. Create a custom field for customer in your Oracle NetSuite instance. See [Creating Custom Entity Fields](#).
 - Specify the value for **Label** as QBCustomerId.
 - Check the **SALE** box in the Applies To section.
 - Check the **Main** box in the Display section.
 - From the **Save** drop-down list, select **Save & Apply to Forms**.

Similarly, create custom fields for item named QuickBookItemId, QuickBookItemName.

5. Configure saved search.
 - a. Go to List, click Search and select Saved Searches.
 - b. Create a new search or use an existing saved search based on your business use case. See [Defining Saved Search](#).
 - c. In the new saved search, select **Invoice** as record type.
 - d. Specify the search title and configure a search criteria that matches the following conditions: QuickBookSyncStatus is **empty**, Type is **Invoice**, and Main Line is **True**. See [Advanced Search Criteria Filters](#).
 - e. Click **Submit**.

Configure QuickBooks

To access the QuickBooks instance from Oracle Integration, you'll need to create a user account for Oracle Integration on QuickBooks and make a few other configurations. See *Prerequisites for Creating a Connection in Using the QuickBooks Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle NetSuite Connection

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

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, https://webservices.netsuite.com/wsd/v2023_1_0/netsuite.wsdl . See Configure Oracle NetSuite .

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

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See <i>Create an Integration Record for Oracle Integration in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See <i>Create an Access Token for the User Account in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier. See <i>Make a Note of the NetSuite Account ID in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

Note

You must enter this information in capital letters.

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

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

A message confirms if your test is successful.

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

Configure the QuickBooks Connection

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

Field	Information to Enter
QuickBooks Environment	Select Production .

- In the Security section, enter the following details:

Field	Information to Enter
Client id	Enter the Client ID obtained when you configured QuickBooks. See Configure QuickBooks .
Client secret	Enter the Client Secret obtained when you configured QuickBooks.

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

You'll now be redirected to the QuickBooks sign in page.

- Enter your QuickBooks account credentials.

A page is displayed asking for permission to interact with the account.

- Click **Yes**.

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the **NS_CUSTOMCONFIG** lookup table to configure the custom script ID.

- In the Lookups section, click the lookup name.
- Enter the value for the following key.

Key	Value
CustomScriptId	custbody_qb_sync_status

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

Activate and Run the Recipe

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

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In your QuickBooks instance, check for new sales receipts.
 - a. To view the list of customers, from the QuickBooks homepage, select **Sales** and choose **Customers**.
 - b. In the customer detailed view, you can view the sales receipts.

Related Documentation

- [Using the Oracle NetSuite Adapter with Oracle Integration 3](#)
- [Using the QuickBooks Adapter with Oracle Integration 3](#)

Create Salesforce Accounts for SAP ERP Customers

Use this recipe to create Salesforce accounts for SAP ERP customers.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates an account in Salesforce whenever a customer record is created in SAP ERP.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a customer-create event is generated in the SAP instance, the integration flow of the recipe is triggered, and it creates a corresponding account in the

Salesforce instance. Basic data such as account name, account number, billing city, billing country, and postal code are synchronized between the SAP ERP and Salesforce platforms.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- An SAP logon client
- An SAP program ID
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SAP ERP and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and transfer customer records between them.

Configure SAP ERP

To configure SAP ERP, see Prerequisites for Creating a Connection in *Using the SAP Adapter with Oracle Integration 3*.

Configure Salesforce

To access Salesforce from Oracle Integration and create accounts for SAP ERP customer records, you must perform certain configurations on your Salesforce instance.

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

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

① Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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. In the **Existing Profile** field, select **Standard User**.
 - ii. Enter a name for the new profile, for example, API Enabled, and click **Save**.

The new profile is now saved, and the Profile Detail page of the new profile is displayed.

- d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

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

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.

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

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the **Quick Links** section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.

A new security token is sent to the email address associated with the account. Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the SAP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, upload the JCO connection properties file.

You must upload the following type of JCO file:

 - – Adapter inbound properties file: To configure the SAP Adapter for both trigger and invoke connections (for example, the Adapter_inbound_Direct.properties file).
3. In the Security section, enter the username and password to connect to the SAP instance.
4. In the Agent Group section, configure an agent group for accessing the service hosted on your premises, behind the firewall.
 - a. Click **Configure Agents**.

The Select an Agent Group page appears.
 - b. Click the name of the agent group.
 - c. Click **Use**.

To configure an agent group, you must download and install the on-premises connectivity agent. See *Downloading and Running the On-Premises Agent Installer* in *Using Integrations in Oracle Integration 3*.

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

A message confirms if your test is successful.

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

Configure the Salesforce Connection

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

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

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce.

Note

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the lookup table, and enter an email ID to receive error messages.

- In the Lookups section, click the lookup name.
- Against the **EmailTo** cell, enter an email ID of your choice to receive error messages.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must push a customer record to the SAP ERP instance, which requires triggering a DEBMAS IDoc in the SAP ERP instance.

- a. In the SAP ERP system, in the **tcode** field, enter the following code: /nBD12, and then click **Enter**.
- b. Select a customer record.
 - i. In the **Customer** field, click the **Search** icon. The Customer Number window appears.
 - ii. In the window, click **Start Search**, and select the required customer record from the list of available records.
- c. Select the output type as **Customer master data distribution**.
 - i. Click the **Output type** field, and then click the **Search** icon.
 - ii. From the list of message types, select the message type and program ID used in the SAP ERP trigger configuration.
- d. Select the configured logical system.
 - i. Click the **Logical system** field, and then click the **Search** icon.
 - ii. From the list, select the configured logical system.

Note

Each logical system has a certain number of customers configured. To see the list of variants of the configured systems, in the SAP ERP system UI, navigate to **More > Goto > Variants > Get**.

- e. After you've configured the parameters, click the **Execute** icon to push the customer record. An information dialog appears.
- f. In the dialog, click **Continue**. Click **Continue** again, if prompted, to continue execution.
- g. To check the status of the execution:
 - i. In the **tcode** field, enter the following tcode: /Nwe02, and then click **Enter**.
 - ii. Click the **Execute** icon to view the status.

Note

Status 3 represents a successful execution, and **Status 29** represents a failed execution.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Salesforce instance and check for the new accounts created.
 - a. On the Salesforce Setup page, click the **Accounts** tab.
 - b. On the Accounts Home page, select **All Accounts** in the **View** field. Click **Go** if necessary.

The list of accounts created are displayed.
 - c. To view an account record, click its name.

Related Documentation

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

Create Salesforce Accounts for SAP S/4HANA Customers

Use this recipe to create corresponding accounts in Salesforce for customer records in SAP S/4HANA.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates corresponding accounts in Salesforce for customer records in SAP S/4HANA as per a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the customer records in the SAP S/4HANA instance and, in turn, creates corresponding accounts in your Salesforce instance. Basic data such as account name, account number, billing city, billing state, billing country, and postal code are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- SAP S/4HANA
- An account on SAP S/4HANA with the Administrator role
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SAP S/4HANA and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and create accounts in Salesforce for S/4HANA customer records.

Configure SAP S/4HANA

Create a communication user and obtain the username and password for the communication user.

1. Log in to the SAP S/4HANA Cloud application with administrator credentials.
2. Search for Communication Management in the search box in the upper right pane, then click **Maintain Communication Users** in the search results.
3. Click **New** to create a new communication user.
4. On the Create Communication User page, enter a user name, description, and password. You can click **Propose Password** to get a system-generated password.
5. Copy the user name and password for your communication user.

You'll need the username and password while configuring your SAP S/4HANA connection from Oracle Integration.

6. Click **Create**.

Configure Salesforce

To access Salesforce using Oracle Integration and create accounts for SAP S/4HANA customer records, you must perform certain configurations 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.

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

① Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

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

The new profile is now saved, and the Profile Detail page of the new profile is displayed.

- d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

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

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user, for example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field.
Note this 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 the same.
Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lighting Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account. Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the SAP S/4HANA Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the host name of the SAP S/4HANA instance.
3. In the Security section, enter the following details:

Field	Information to Enter
Username	Enter the username of the SAP S/4HANA instance. See Configure SAP S/4HANA .
Password	Enter the password of the SAP S/4HANA instance.

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

A message confirms if your test is successful.

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

Configure the Salesforce Connection

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

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

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce.

Note

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the lookup table to map Salesforce country codes to S/4HANA country codes. In addition, enter an email ID to receive error messages.

- In the Lookups section, click the lookup name.
- Against the **EmailTo** cell, enter an email ID of your choice to receive error messages.
- In the second and third **Value** columns, map the country codes between the Salesforce and S/4HANA instances.
- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the lastRun parameter. Update the parameters with appropriate values.

The lastRun parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the S/4HANA customer records created 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. To fetch all the customer records from the beginning, set the value as zero.

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Salesforce instance and check for the new accounts created.
 - a. On the Salesforce Setup page, click the **Accounts** tab.
 - b. On the Accounts Home page, select **All Accounts** in the **View** field. Click **Go** if necessary. The list of accounts created is displayed.
 - c. To view an account record, click its name.

Related Documentation

- [Using the SAP S/4HANA Cloud Adapter with Oracle Integration 3](#)
- [Using the Salesforce Adapter with Oracle Integration 3](#)

Create Salesforce Cases for Jira Issues

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)

- [Activate and Run the Recipe](#)

About This Recipe

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

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.

Create an API Token in Your Atlassian Account

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

To create an API token:

1. 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.
4. 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**.
6. 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.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.

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

Note

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

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.

The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.

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

Note

If you're shown the Lightning 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. 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:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Cases**, and then click **Fields**.
 - b. On the Case Fields page, scroll to the Case Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the REST Jira Connection

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

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

3. 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 .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Salesforce Connection

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

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

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

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update integration property for the integration flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration property with appropriate value.
 - **emailNotification:** This integration property holds the email address to which notifications of errors in the integration's execution are sent.
 - c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, 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. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Salesforce instance and check for the new (or updated) case records.
 - a. On the Salesforce Setup page, click the **Cases** tab.

Note

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**.

- b. 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 3](#)
- [Using the Salesforce Adapter with Oracle Integration 3](#)

Create SAP ERP Orders for Closed Salesforce Opportunities

Use this recipe to create SAP ERP orders for Salesforce opportunities won.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates an order in SAP ERP whenever an opportunity record closes with the status *Closed Won* in Salesforce. It uses the standard Salesforce REST Adapter and SAP Adapter.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a Salesforce opportunity record closes with the status *Closed Won*, the integration flow of the recipe is triggered, and it creates a corresponding order in the SAP ERP instance. Basic data such as opportunity name, closed date, stage, account name, amount, and probability are synchronized between the Salesforce and SAP ERP platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- An SAP logon client
- An SAP program ID
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and SAP ERP instances in order to successfully connect to these external systems using Oracle Integration and transfer customer records between them.

Configure Salesforce

To configure Salesforce, see Prerequisites for Creating a Connection in *Using the Salesforce REST Adapter with Oracle Integration 3*.

Configure SAP ERP

To configure SAP ERP, see Prerequisites for Creating a Connection in *Using the SAP Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Salesforce REST Connection

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

Field	Information to Enter
Salesforce Domain URL	Specify the Salesforce domain URL, see Configure Salesforce
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

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

Field	Information to Enter
Security Policy	Select Authorization Code Credentials .
Client ID	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .

Field	Information to Enter
Client Secret	Enter the password of the account created for Oracle Integration on Salesforce.

Note

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **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 Salesforce login page.
7. Enter your Salesforce account credentials and click **Continue** to log in.
You are now displayed a page that prompts you to authorize the connection to the Salesforce application created previously.
8. On the Salesforce web page, click **ACCEPT**.
You're informed that access to Salesforce is allowed. You can now switch back to the Salesforce Connection window of Oracle Integration to test your connection.
9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
10. To return to the project workspace, click **Go back** .

Configure the SAP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, upload the JCO connection properties file.
You must upload the following type of JCO file:
 - – Adapter inbound properties file: To configure the SAP Adapter for both trigger and invoke connections (for example, the Adapter_inbound_Direct.properties file).
3. In the Security section, enter the username and password to connect to the SAP instance.
4. In the Agent Group section, configure an agent group for accessing the service hosted on your premises, behind the firewall.
 - a. Click **Configure Agents**.
The Select an Agent Group page appears.
 - b. Click the name of the agent group.
 - c. Click **Use**.
To configure an agent group, you must download and install the on-premises connectivity agent. See Downloading and Running the On-Premises Agent Installer in *Using Integrations in Oracle Integration 3*.

5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Update Integration Property

Update the integration property with appropriate value.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the following integration property with appropriate value.
 - **Email_Id**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
3. Click **Submit**.

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

Configure the Lookup Table

Edit the lookup table, and enter an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. Label the first and second column as SAP_Parameter and Parameter_Value respectively.
3. In the SAP_Parameter column, enter the parameter name as Convert.
4. In the Parameter_Value column, enter the value X.
5. Click **Save**. If prompted, click **Save** again.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page, click the **Opportunities** tab.
 - c. On the Opportunities Home page, select **All Opportunities** in the **View** field and click **Go!**.
All the opportunities in your Salesforce instance are displayed.
 - d. Find a sample opportunity record that hasn't been closed, and click its name to open it.
 - e. On the opportunity's page:

- i. 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 have now successfully triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your SAP ERP instance and check for the record created.
 - a. In the SAP ERP system, in the **tcode** field, enter the following code: /Nva02, and then click **Enter**.
 - b. In the resulting window, enter the order number in the **Order** field to view the order details.

Note that the order number is the Sales Document Number copied from your Oracle Integration instance.

Related Documentation

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

Create Suppliers in SAP Ariba for Vendors in SAP ERP

Use this recipe to create corresponding supplier records in SAP Ariba when vendor records are created in SAP ERP.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a supplier in SAP Ariba whenever a vendor is created in SAP ERP.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a vendor record is created in the SAP ERP instance, the integration flow of the recipe is triggered, and it creates a corresponding supplier record in SAP Ariba. Basic data, such as contact, contact ID, email address, fax, name, and postal address, are synchronized between the SAP ERP and SAP Ariba platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- An SAP logon client
- An SAP program ID
- An SAP Ariba instance
- An account on SAP Ariba with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SAP ERP and SAP Ariba instances in order to successfully connect to these external systems using Oracle Integration and transfer records between them.

Configure SAP ERP

To configure SAP ERP, see Prerequisites for Creating a Connection in *Using the SAP Adapter with Oracle Integration 3*.

Configure SAP Ariba

To configure SAP Ariba, see Complete Ariba SOAP Web Services Prerequisites in *Using the SAP Ariba Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the SAP ERP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, upload the JCO connection properties file.
You must upload the following type of JCO file:
 - Adapter inbound properties file: To configure the SAP Adapter for both trigger and invoke connections (for example, the Adapter_inbound_Direct.properties file).
3. In the Security section, enter the username and password to connect to the SAP ERP instance.
4. In the Agent Group section, configure an agent group for accessing the service hosted on your premises behind the firewall.
 - a. Click **Configure Agents**. The Select an Agent Group page appears.
 - b. Click the name of the agent group.
 - c. Click **Use**.
To configure an agent group, you must download and install the on-premises connectivity agent. See Download and Run the Connectivity Agent Installer in *Using Integrations in Oracle Integration 3*.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

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

Configure the Oracle SAP Ariba Connection

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

Field	Information to Enter
Host Name	Enter the SAP Ariba host name. To obtain the host name, see Configure SAP Ariba .
Partition (Site Name)	Enter the SAP Ariba partition.

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

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username created on SAP Ariba during the endpoint configuration. See Configure SAP Ariba .
Password	Enter the password created on SAP Ariba during the endpoint configuration.

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the lookup table and enter an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. In the **Property_Name** column, enter the property name as **EmailTo**.
3. In the **Property_Value** column, enter an email ID of your choice.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must perform the following tasks in the SAP ERP instance.

- a. Check the Oracle Integration connection status.
 - i. In the **tcode** field, enter /nsmgw, and then click **Enter**.
 - ii. Click **Goto**, then select **Logged on Clients**. A list of logged-on clients appears.

- iii. In the list, check if the configured server address appears.
 - b. Test the connection.
 - i. In the **tcode** field, enter /nsm59, and then click **Enter**.
 - ii. From the list of RFC Connections, expand **TCP/IP Connections**, and find your program ID.
 - iii. Select the program ID, and click **Connection Test**.
 - c. Push a vendor record to SAP ERP.
 - i. In the **tcode** field, enter /nBD14, and then click **Enter**.
 - ii. Select a vendor account number.
 - In the **Account Number** field, click **Search** . The Account Number window appears.
 - In the window, click **Start Search** , and select the required vendor record from the list of records displayed.
 - iii. Select the **Message Type** as **CREMAS**.
 - iv. Select the Target System.
 - v. After you've configured the parameters, click **Execute**  to push the vendor record. An information dialog appears.
 - vi. In the dialog, click  to continue. Click  again, if prompted, to continue execution.
 - d. Check the status of the execution.
 - i. In the **tcode** field, enter /nwe02, and click **Enter**.
 - ii. In the IDoc List window, click **Execute**  to view the status.

 Note

Status 3 represents a successful execution, and **Status 29** represents a failed execution.

- 3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
- 4. Log in to your SAP Ariba instance and check for the new supplier record created.
 - a. On the SAP Ariba home page, click **Home**, then **Common Actions**, and then **Core Administration**.
 - b. On the **Administration** tab, click **Supplier Manager**.
 - c. In the Supplier Manager window, click **Suppliers**.
 - d. Enter the contact information in the **Contact** field, and click **Search**.
 - e. In the list of records displayed, find the required supplier record, and click its name to view it.

Related Documentation

- *Using the SAP Adapter with Oracle Integration 3*
- *Using the SAP Ariba Adapter with Oracle Integration 3*

Create Users in OCI Identity and Access Management for Persons in PeopleSoft

Use this recipe to create a user in Oracle Cloud Infrastructure Identity and Access Management (OCI Identity and Access Management) when a person is added in PeopleSoft.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates corresponding users in OCI Identity and Access Management for persons added in PeopleSoft.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a new person is added in PeopleSoft, the integration flow of the recipe gets triggered and it subsequently creates a corresponding user in OCI Identity and Access Management. Thus user account data is synchronized between PeopleSoft and OCI Identity and Access Management.

System and Access Requirements

- Oracle Integration 3
- PeopleSoft
- An account on PeopleSoft with the Administrator role
- OCI Identity and Access Management
- An account on OCI Identity and Access Management with the Administrator role

Before You Install the Recipe

To successfully connect to PeopleSoft using Oracle Integration and synchronize user accounts between PeopleSoft and OCI Identity and Access Management, you must perform certain configuration tasks on your PeopleSoft instance.

1. Define integration gateways and load connectors.
 - a. Configure gateway URL and activate domain.
 - i. Go to **PeopleTools > Integration Broker > Configuration > Integration Gateways**.
 - ii. Leave the default gateway URL as is or enter the one you want to use.
 - iii. Click **Ping Gateway**.
Ensure that a message displays that the gateway is active.

- iv. To process asynchronous messages, one application server domain must be active. In the Integration Broker Domains pane, activate the appropriate domain you want to use.
 - v. Click **Save**.
- b. Load gateway connectors.
 - i. Go to **PeopleTools > Integration Broker > Configuration > Integration Gateways**.
 - ii. Search the existing gateways. If no gateway satisfies your search criteria, click the **Add A New Value** link.
 - iii. In the **Integration Gateway ID** field, enter the integration gateway ID, for example: IAM_LOCAL, and click **Add**.
 - iv. Select the **Local Gateway** check box, and in the **URL** field enter the gateway URL that you configured in the previous step.
 - v. Click **Load Gateway Connectors**. All loaded gateway connectors are displayed in the **Connectors** area.
 - vi. Click **Save** to save the gateway information.
- 2. Configure integration gateway properties.
 - a. Go to **PeopleTools > Integration Broker > Configuration > Integration Gateways**.
 - b. Search the gateway you defined in step 1b.
 - c. Open the gateway and click the **Gateways Setup Properties** link next to the integration gateway URL field.
 - d. In the displayed Gateway Properties Sign-in page, enter **User ID** and **Password**, and click **OK**.
 - e. In the displayed PeopleSoft Node Configuration page, click the **Advanced Properties Page** link.
 - f. Configure gateway properties in the displayed editor.

The following integration gateway properties have to be configured:

Delivered Connector Configuration

 - `ig.isc.serverURL=PeopleSoft_server_name:port ig.isc.userid=username ig.isc.password=password ig.isc.toolsRel=PeopleTools_version`
 - Use the supplied Password Encryption Utility to generate an encrypted password for `ig.isc.password`.
 - g. Click **OK** to save the properties.
 - h. Click **Save** to save the integration gateway configuration.
- 3. Set up integration.
 - a. Define nodes.

Nodes represent any organization, application or system that will play a part in integrations. For example, nodes can represent customers, business units, suppliers, other trading partners, external or third-party software systems, and so on.

 - i. Go to **PeopleTools > Integration Broker > Integration Setup > Node Definitions**.
 - ii. Search the existing nodes. If no node satisfies your search criteria, click the **Add A New Value** link. Enter **Node Name** and click **Add**.

iii. Click the **Node Definitions** tab to define the node:

- In the **Description** field, enter a description.
- From the **Node Type** drop-down field, select **PIA**.
- Enter **Default User ID**.
- Select the **Active Node** checkbox.

iv. Click the **Connectors** tab and define the connectors.

- Specify **Gateway ID** or click the button to look up the Gateway ID you defined in step 1b.
- Specify **Connector ID** or click the button to look up the Connector ID you defined in step 1b.

The connector properties are loaded as shown in the following table.

Field	Action/Description
HEADER	Property Name: sendUncompressed Required: Enable this option. Value: Y Data Type: STRING Description: A descriptive text entry is optional.
HEADER	Property Name: Content-Type Value: text/xml Data Type: STRING Description: A descriptive text entry is optional.
HTTPPROPERTY	Property Name: Method Required: Enable this option. Value: POST Data Type: STRING Description: A descriptive text entry is optional.
PRIMARYURL	Property Name: URL Required: Enable this option. Value: The URL that points to the integration server. For example: https://IntegrationServer_Host:Port/ic/api/integration/v2/flows/rest/project/ORCL-R-PEOPLESOFT_IAM_USER_SYNC/PEOPLESOFT-IAM-USER-SYNC/1.0/persondetails Data Type: STRING Description: A descriptive text entry is optional.

Field	Action/Description
HEADER	Property Name: Authorization Required: Enable this option. Value: The user name and password used as authorization to start the integration server. Use the following format, along with the encoding flow service listed below, to encode the user name and password: "Basic" (+ Base64Encode service using username + ":" + password) To encode the username and password, use the following utility flow service: <code>wm.adapter.peoplesoft.outbound.utils.encoding</code> where the input for the encoding service is a string constructed from the username and password, as follows: "username"+" ":"password". For example, if the username is "name" and the password is "pwd", the input for the encoding service would be "name:pwd". The result of encoding flow service for these values would be "bmFtZTpwd2Q". Thus, the value of the Authorization field would be "Basic bmFtZTpwd2Q". Data Type: Leave this field blank. Description: A descriptive text entry is optional.

- Click **Ping Node**. Ensure the result is success.
- Click **Return**.
- Click **Save**.

The specified node is created.

b. Define messages.

Message definitions provide the physical description of the data that is being sent, including fields, field types, and field lengths.

- Go to **PeopleTools > Integration Broker > Integration Setup > Message Definitions**.
- Search the existing messages. If no message satisfies your search criteria, click the **Add A New Value** link.
- In the Add New Message window:
 - From the **Type** drop-down list, select **Rowset**.
 - In the **Message Name** field, enter a name for the message. For example: COUNTRY_SYNC.
 - In the Message Version field, enter the version. For example: VERSION_1
 - Click **Add**.
- Click the **Message Definition** tab and define the message.
 - Enter the message's description in the **Description** field.
 - From the **Owner ID** drop-down list, select **Enterprise Components**.
 - Click the **Add Record To Root** link to add a level 0 record to the root node.
 - Click the record link, you can also add more records to the level 0 record or its child records.
 - Click **Save**.

The specified message is created.

c. Define message queues.

A message queue isolates different groups of service operations from each other. It needs to be created exclusively to handle messages meant for the plug-in. You can also choose an existing message queue. If a message does not exist for a record from which data is to be exported, it needs to be created. Modifications to the existing messages might be required.

d. Define services.

PeopleSoft Integration Broker enables you to take an existing component interface and create a service which can be used to invoke the component interface.

- i. Go to **PeopleTools > Integration Broker > Integration Setup > Service Definitions**.
- ii. Search the existing services. If no service satisfies your search criteria, click the **Add A New Value** link.
- iii. Enter the service name, and then click the **Add** button.
- iv. In the Services window, fill in the service information.
 - Enter a description for the service in the **Description** field.
 - From the **Object Owner ID** drop-down field, select **Enterprise Components**.
 - In the Service Operations area, define the service operation. Enter the **Service Operation**, select the type of operation from the **Operation Type** drop-down list, and then click **Add**.
To define service operation, follow step 3e.
- v. Click **Save**.

The specified service is created.

e. Define service operations.

A service operation definition consists of general information about an operation, such as its name, description, and so on. It also specifies an operation type, which determines how the operation is to be processed, synchronously or asynchronously. In addition, it contains routings which determine the direction, inbound or outbound, of the service operation. A service operation has one or more handlers, which contain and run the programming logic for sending or receiving the message, manipulating message content, and other functions.

With Integration Broker (IB), you can define and customize service operations. The steps to define a service operation are the same for all IB activities. However, based on the IB activity to be used, you need to select the appropriate operation type from the following:

Asynchronous - One Way: for IBRequest Activity.

To define a service operation:

- i. Go to **PeopleTools > Integration Broker > Integration Setup > Service Operation Definitions**.
- ii. Search the existing service operations. If no service operation satisfies your search criteria, click the **Add A New Value** link.
- iii. In the Add New Service Operation pane, enter the following details:
 - Specify the service or click the button to look up the service you defined in step 3d.

- Specify the **Service Operation** name.
 - Select the appropriate **Operation Type** from the drop-down list.
 - Click **Add**.
- iv. Click the **General** tab and enter the following details:
- Enter **Operation Definition**.
 - From the **Object Owner ID** drop-down list, select **Enterprise Components**.
 - Enter **Version**.
 - Select the **Active** checkbox.
 - Specify **Message.Version** or click the button to look up the message you defined in step 3b.
 - Specify **Queue Name** or click the button to look up the message queue you defined in step 3c.
- v. Click the **Routing** tab. Enter **Routing Name** and click **Add**.
- vi. Define the routing by following step 3f.
- vii. To choose the routing you defined and want to use, select the **Selected** check box.
- viii. Click **Save**.

The specified service operation is created.

f. Define routings.

Routing definitions determine the sender and receiver of an integration. With routing definitions, you can specify inbound and outbound transformations that enable you to transform data structures into those that the sending or receiving systems can understand.

With Integration Broker, you can define and customize routings. The steps to define a routing are the same for all IB activities. However, based on the IB activity to be used, you need to select the appropriate sender node and receiver node.

To define a routing:

- i. Go to **PeopleTools > Integration Broker > Integration Setup > Routing Definitions**.
- ii. Search the existing routings. If no routing satisfies your search criteria, click the **Add A New Value** link.
- iii. Enter **Routing Name** and click **Add**.
- iv. Click the **Routing Definitions** tab and define the routing:
 - Specify **Service Operation** or click the button to look up the service you defined in step 3e.
 - Enter **Version**.
 - Enter **Description**.
 - Specify **Sender Node** or click the button to look up the node you defined in step 3a.
 - Specify Receiver Node or click the button to look up the node you defined in step 3a.
 - From the **Object Owner ID** drop-down list, select **PeopleTools**.
- v. Click the **Connector Properties** tab and enter the following information:

- Specify **Gateway ID** or click the button to look up the gateway you defined in step 1.
- Specify **Connector ID** or click the button to look up the connector you defined in step 2.

vi. Click **Save**.

The specified routing is created.

① Note

You can get the schema payload from PeopleSoft Person Node. It should be similar to the XML payload in the following file: [PersonBasicSync.xml](#). Ensure that the XML payload matches with the Oracle Integration REST Trigger request.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the IAM REST Connection

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

Field	Information to Enter
Connection type	Select REST API Base URL .
Connection URL	Specify the endpoint URL to use. For example: <code>https://idcs-825cxxxxxxx.identity.oraclecloud.com:443</code>

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

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Enter the URL from which to obtain the access token.
Client Id	The client identifier issued to the client during the registration process.
Client Secret	Enter the client secret.
Scope	Enter the scope of the access request. For example: <code>urn:opc:idm:__myscopes__</code>
Auth Request Media Type	Enter the format of the data you want to receive. For example: <code>application/x-www-form-urlencoded</code>
Client Authentication	Send client credentials as basic auth header: Pass the client ID and client secret in the header as basic authentication.

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

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
To run the recipe, log in to your PeopleSoft instance and add a person. This triggers the **Oracle PeopleSoft IAM User Sync** integration flow, and the integration now creates a corresponding user in OCI Identity and Access Management.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In OCI Identity and Access Management, check if a new user corresponding to the person added in PeopleSoft has been created.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)
- [Integrating Person Data \(PeopleSoft\)](#)
- [OCI Identity and Access Management documentation](#)

Create Workers in Oracle HCM Cloud

Use this recipe to create workers in Oracle HCM Cloud based on hiring data from an external system.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports a file with worker records from an FTP server and creates corresponding workers in Oracle HCM Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When triggered, the integration flow of the recipe reads a file that contains data for new workers from an external system on an FTP server. Then, it transforms and formats the file in Oracle Integration to generate the REST payload. Finally, it creates different worker types in Oracle HCM Cloud. Further, it stores the successfully processed worker records and records that failed during processing in respective folders in the FTP server.

System and Access Requirements

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

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud instance and FTP server in order to successfully connect to these external systems using Oracle Integration and create workers in Oracle HCM Cloud.

Configure Oracle HCM Cloud

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

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

1. Create a user account for Oracle Integration. Note the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle HCM Cloud from Oracle Integration.
2. Assign required roles to the user account.

See Assign Required Roles to an Integration User in *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Access Your FTP Server and Create Folders

Obtain an SFTP server, ensure that you're able to access it, and create required folders.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create the following folders on the FTP server.
 - InputFolder: This folder is used to place the input files with the worker records from an external system. The input files will be processed in Oracle Integration.

Note that for successfully processing the worker record files in Oracle Integration, the input files should be in a particular format. For the purpose of testing, you can download a sample input file `Employee_2019-12-20T13h00.json` from [Sample file](#).

- **ResultFolder:** This folder is used to store the result of the successfully processed worker records.
- **ArchiveFolder:** This folder is used to store the processed input files. Once they are processed, input files from the `InputFolder` are moved into this folder.
- **ErrorFolder:** This folder is used to keep the files with records that failed to process.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle HCM Cloud

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

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

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

A message confirms if your test is successful.

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

Configure the FTP Connection

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

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

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

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the lookup table to map country names to territory codes.

1. In the Lookups section, click the lookup name.
2. Map the values under the **CountryName** and **TerritoryCode** lookup keys.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **InputFolder**: This integration property holds the path of the InputFolder that you created in your FTP server. See [Access Your FTP Server and Create Folders](#).
 - b. **ArchiveFolder**: This integration property holds the path of the ArchiveFolder that you created in your FTP server.
 - c. **ErrorFolder**: This integration property holds the path of the ErrorFolder that you created in your FTP server.
 - d. **ResultFolder**: This integration property holds the path of the ResultFolder that you created in your FTP server.
 - e. **FromAddress**: This integration property holds the email address from which error notification emails are sent. Enter an email address of your choice.
 - f. **ToAddress**: This integration property holds the email address to which error notification emails are sent. Enter an email address of your choice.
3. Click **Submit**.

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

Activate and Run the Recipe

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

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle HCM Cloud instance, and check if corresponding workers are created for the worker records imported from FTP.

Related Documentation

- [Using the Oracle HCM Cloud Adapter with Oracle Integration 3](#)
- [Using the FTP Adapter with Oracle Integration 3](#)

Create Zendesk Tickets for Jira Issues

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates a ticket in Zendesk for each Jira issue as per a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe 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.

Note

- 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 24.04 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

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.

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:

1. 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.
4. 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**.
6. 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 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the REST Jira Connection

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

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

3. 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 .

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

A message confirms if your test is successful.

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

Configure the REST Zendesk Connection

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

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

3. 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.

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

A message confirms if your test is successful.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration properties of the integration flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.
 - **emailNotification**: This integration property holds the email address to which notifications of errors in the integration's execution are sent.
 - **FromEmailAddress**: This integration property holds the email address from which notifications of errors in the integration's execution are sent.

Note

You can configure the email addresses of the senders and recipients of notification errors from the Notifications page. To access the Notifications page, click **Settings** in the navigation menu, and then click **Notifications**.

- c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run 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.

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your 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 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)

Create Zendesk Tickets for Salesforce Cases

Use this recipe to create Zendesk tickets for cases created in Salesforce.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a ticket in Zendesk whenever a case is created in Salesforce.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a case is created in Salesforce, the integration flow of the recipe is triggered, and it creates a corresponding ticket in the Zendesk instance. Basic data such as subject, description, and priority are synchronized between the Salesforce and Zendesk platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Zendesk
- An account on Zendesk with the Administrator role

Before You Install the Recipe

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

Configure Salesforce

To access Salesforce from Oracle Integration and read case records, you must perform certain configurations on your Salesforce instance.

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

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce case is created.

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

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

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

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.

The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.

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

Note

If you're shown the Lightning 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.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.

- c. On the New Outbound Message page:
 - i. Select **Case** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the message.
 - In the **Endpoint URL** field, enter the following URL: `https://<OIC_FQDN:OIC_SSL_PORT>/ic/ws/integration/v1/flows/salesforce/ORCL-R-SFDC_CASETOZEN_TICKET_SYN/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 **Account 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 case is created.
 - 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 the rule's name.
 - Optionally, enter a description for the rule.
 - In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
 - In the Rule Criteria section, select the **formula evaluates to true** option, and define the following rule criteria:

(Case: Number NOT EQUAL TO null)
 - Click **Save & Next**.
 - iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
 - iv. On the Select Existing Actions page:
 - Select **Outbound Message** in the **Search** field.

- In the **Available Actions** box, select the outbound message you created previously, for example,

Outbound Message: <name of your outbound message>

- Click **Add** to move your selection to the **Selected Actions** box.
- Click **Save**.

- v. Click **Done**.

The new workflow rule is created and displayed on your browser.

- d. Click **Activate** to activate the rule.

Configure Zendesk

To access Zendesk from Oracle Integration and create tickets, you'll need to create an OAuth client on your Zendesk instance. See *Create an OAuth Client in Using the Zendesk Adapter with Oracle Integration 3*.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Salesforce Connection

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

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

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

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

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

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

A message confirms if your test is successful.

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

Configure the Zendesk Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Zendesk host name.
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 Zendesk. See Configure Zendesk .
Client Secret	Enter the client secret of the OAuth application created in Zendesk.
Scope	Enter write.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.
You'll now be redirected to the Zendesk login page.
7. Enter your Zendesk account credentials and click **Sign in**.
8. After you see an access allowed message, switch back to the Zendesk Connection window of Oracle Integration to test your connection.
9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
10. To return to the project workspace, click **Go back** .

Configure the Lookup Table

The lookup table of the recipe contains a mapping of Salesforce priority codes to Zendesk priority codes. Additionally, it contains a field where you can enter an email ID on which you'd like to receive error messages. Edit the lookup table to update any of these values.

1. In the Lookups section, click the lookup name.
2. In the **Salesforce** or **Zendesk** columns, edit or add priority values of the respective systems.
3. In the **EmailTo** row, enter an email ID of your choice.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page (in the Classic UI), click the **Cases** tab.
 - c. On the Cases Home page, click **New**.
 - d. On the New Case page, enter the case details, such as subject, description, and priority. Click **Save**.

You've now successfully triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Zendesk instance and check for the new ticket created.

On the Zendesk home page, search for the required ticket.

Related Documentation

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

Delete Old Files

Use this recipe to delete old files from File Server in Oracle Integration.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe deletes files that are older than the value specified in the Minimum Age parameter from File Server. To use the recipe, you must install the recipe and configure the connections and other resources within it.

When triggered, the integration flow performs the following tasks:

- Based on the lookup key values and the value specified in the Minimum Age parameter configured in the List Files operation, lists files that are older than the minimum age specified. The Minimum Age parameter can be configured to accept values in seconds such as 7776000 s (90 days), 5184000 s (60 days), or 2592000 s (30 days).
- Uses the for-each action to iterate through each of the files in the List Files response, and deletes files using the Delete a File operation.

Note

In a single integration run only 100 files can be deleted.

- In case of any error, sends notifications to the email Id configured in the **ToEmailAddress** lookup key.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- File Server
- An SFTP client

Before You Install the Recipe

To use File Server, you must enable it for your Oracle Integration instance. See [Enable File Server in *Using File Server in Oracle Integration 3*](#).

1. Log in to the server using your user name and password through an SFTP client; for example, FileZilla.
2. Create a directory on the server from which to purge files. Note down the path of the directory.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the FTP Connection

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

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

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

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

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the **ORCL-OIC_SFTP_PURGE_FILES** lookup table and enter appropriate values for the look up keys.

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

Key	Description
IntegrationID	The key to fetch all the remaining values from the lookup. For example: ORCL - OIC_SFTP_PURGE_FILES
FromEmailAddress	Specifies the email address to which the error notifications are sent. Enter an email address of your choice.
ToEmailAddress	Specifies the email address from which the error notifications are sent. Enter an email address of your choice.
EmailSubject	Specifies the subject of the email error notification.
OICSFTPPath	The FTP directory path of the location from which to purge files. For example: /home/users/In
FilePrefix	The prefix of the files that are to be purged. For example: *.*

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

Activate and Run the Recipe

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

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your SFTP server and check if files older than the value that was specified in the Minimum Age parameter of List File operation have been deleted.

Related Documentation

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

Encrypt and Decrypt Data Using OCI Vault

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

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

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

System and Access Requirements

- Oracle Integration, Version 22.1.3.0.0 or higher
- OCI Vault
- An account on OCI with the Administrator role

Before You Install the Recipe

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

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

1. Create an OCI Vault for your master encryption keys. See [Managing Vaults](#).
2. Create a new master encryption key. See [Managing Keys](#).

Note

You can also upload your own master encryption keys if you want to manage your encryption keys.

3. Get your Tenancy and User OCIDs. See [Where to Get the Tenancy's OCID and User's OCID](#).
4. Create your own private key or generate it from the Oracle Cloud Console. See [How to Generate an API Signing Key](#).
5. Get the key's fingerprint. See [How to Get the Key's Fingerprint](#).

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle OCI Vault Connection

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

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

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

Field	Information to Enter
Security Policy	Select Select OCI Signature Version 1 .
Tenancy OCID	Enter the Tenancy OCID obtained earlier. See Before You Install the Recipe .
User OCID	Enter the User OCID obtained earlier.
Private Key	Enter the API key generated earlier.

Note

Before you upload the private key, you must convert it into the PKCS1 format.

Finger Print	Enter the key's fingerprint obtained earlier.

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

A message confirms if your test is successful.

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

Configure the Lookup Table

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

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

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

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

Activate and Run the Recipe

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

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

A message confirms that the integration has been activated. Refresh the page to view the updated status of the integration.

2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the plain text or cipher text, which you want to encrypt or decrypt. Provide the text in the POST request's **Body** field. See the subsequent step for example request payloads.

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

3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click **Body** and enter the request data.

Example request payload to encrypt data:

```
{
  "plain_text": "oracle"
}
```

Example request payload to decrypt data:

```
{
  "cipher_text":
  "QW3HfP0AD4bvAFJgNV2RrrLrbuNAmKxa2MhaUhFlsgUU2DwAgMWxp6MWZFcAAAAA"
}
```

c. Click **Run**

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

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

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)
- [Oracle Cloud Infrastructure Documentation](#)

Export Campaign Data in Bulk from Snowflake to Salesforce Marketing Cloud Using GCP Storage

Use this recipe to export campaign data in bulk from Snowflake to Salesforce Marketing Cloud (SFMC) using Google Cloud Platform (GCP) Storage.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe exports campaign data in bulk from Snowflake to Salesforce Marketing Cloud using an external stage (GCP Storage).

To use the recipe, you must install the recipe and configure the connections and other resources within it. You can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow fetches the campaign data from Snowflake and exports it to a GCP Storage bucket in comma separated value (CSV) format. Subsequently, it downloads the CSV file from the GCP Storage bucket and creates campaign objects in Salesforce Marketing Cloud. Basic data such as Name, Description,

CampaignCode, Favorite, Color and so on are synchronized between Snowflake and Salesforce Marketing Cloud.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Snowflake
- An account on Snowflake with the System Administrator role
- GCP bucket that is configured with a storage integration in the Snowflake account as an external stage.
- Salesforce Marketing Cloud
- An account on Salesforce Marketing Cloud with the Administrator role
- GCP Storage
- An account on GCP Storage with the System Administrator role
- Ensure your Snowflake storage integration has access to the GCP bucket by granting permission in the GCP bucket (by adding Principals and assigning the Storage Object Admin role)

Before You Install the Recipe

You must perform the following configuration tasks on your Snowflake, GCP Storage and Salesforce Marketing Cloud instances to successfully connect to these external systems using Oracle Integration and import campaign data.

Configure Snowflake

Complete the following tasks in your Snowflake instance to successfully connect to it from Oracle Integration and export campaigns.

- For general configuration steps, see Prerequisites for Creating a Connection in *Using the Snowflake Adapter with Oracle Integration 3*.
- Ensure the table with column definitions such as Name, Description, CampaignCode, Favorite, and Color is created in Snowflake.
- Ensure that a Cloud Storage Integration is created in Snowflake and grant the Service Account the necessary permissions to access objects within the Cloud Storage bucket. See [Create a Cloud Storage Integration in Snowflake](#).

Configure GCP Storage

To configure GCP Storage, see Prerequisites for Creating a Connection in *Using the GCP Storage Adapter with Oracle Integration 3*.

Configure Salesforce Marketing Cloud

To access Salesforce Marketing Cloud from Oracle Integration and import campaign data, you must perform certain configurations on your Salesforce Marketing Cloud instance.

- For general configuration tasks, see Prerequisites for Creating a Connection in *Using the Salesforce Marketing Cloud Adapter with Oracle Integration 3*.
- To obtain Client ID and Secret, see [OAuth client ID credentials](#).

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Snowflake Connection

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

Field	Information to Enter
Instance URL	URL of your Snowflake account, typically provided by Snowflake when you sign up. See Configure Snowflake .
Warehouse Name	Name of the Snowflake virtual warehouse where the data is stored and processed.
Database Name	Name of the Snowflake database where the desired tables are located.
Schema Name	Name of the schema within the specified database where the tables are located.

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

Field	Information to Enter
Security Policy	Select Authorization Code Credentials .
Client ID	Enter the Client Id of the account created for Oracle Integration on Snowflake. See Configure Snowflake .
Client Secret	Enter the secret value of the account created for Oracle Integration on Snowflake.
Scope (Optional Security)	refresh_token

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. Once the consent is approved, click **Test** in the connection window to ensure that your connection is successfully configured, click **Test** again.

A message confirms if your test is successful.

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

Configure the GCP Storage Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the unique identifier assigned to your GCP project in the **Project Id** field. This ID is used to authenticate requests and access resources within your project.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Authorization Code Credentials .
Google Client ID	Enter the Google Client Id of the account created for Oracle Integration on GCP Storage. See Configure GCP Storage .
Google Client Secret	Enter the secret value associated with the Client ID.
Google Authorization Scope	This refers to the specific permissions or scopes that the OAuth 2.0 credentials (Client ID and Client Secret) have been granted for accessing Google Cloud Storage (GCS). The scope determines what actions the integration can perform, such as read-only access or full access to GCS resources.

- Click **Save**. If prompted, click **Save** again.
- Click **Provide Consent**.
- Once consent is approved, click **Test** on the connection window to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

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

Configure the Salesforce Marketing Cloud Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the environment subdomain (A unique identifier) with which user needs to connect in the **Subdomain** field.
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Server-to-Server Integrations Security Policy .
Client ID	Enter the client Id of the account created for Oracle Integration on SFMC. See Configure Salesforce Marketing Cloud .
Client Secret	Enter the secret value of the account created for Oracle Integration on SFMC.

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

A message confirms if your test is successful.

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

Configure the Lookup Table

Edit the **SNOWFLAKE_SFMC_LOOKUP** lookup table and enter appropriate values.

- In the Lookups section, click the lookup name.
- In the **Property_name** column, enter the property name.

3. In the **Property_value** column, enter the values as shown in the below table to which the property names must be mapped.

Property_name	Property_value
StageName	Specify the stage name associated with the Snowflake endpoint.
SubDirectory	Specify a sub directory within the specified stage.

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration properties of the integration flow. You can update the integration properties for the integration flow only after activating the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Snowflake Salesforce Marketing Cloud Campaign Sync** integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.
 - i. **EmailTo**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
 - ii. **RetryCount**: You can specify the number of retries if any invoke fails at run time. The default value is set 3.
 - iii. **WaitForCampaignCreationRetry**: This integration property controls retries and timeouts for confirming the successful creation of campaigns in Salesforce Marketing Cloud. The default value is set to 60 secs.
 - c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Snowflake Salesforce Marketing Cloud Campaign Sync** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. In your Salesforce Marketing Cloud instance check for the campaign objects created.

- a. Log in to Salesforce Marketing Cloud.
- b. On the Home page, click **Campaigns**.
You can see the list of campaigns created.

Related Documentation

- *Using the Snowflake Adapter with Oracle Integration 3*
- *Using the GCP Storage Adapter with Oracle Integration 3*
- *Using the Salesforce Marketing Cloud Adapter with Oracle Integration 3*

Export Data in Bulk from Oracle ERP Cloud to an FTP Server

Use this recipe to export data in bulk from Oracle ERP Cloud to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe extracts BI Publisher (BIP) report data from Oracle ERP Cloud in an asynchronous manner and uploads the data to an FTP server.

To use the recipe, you must install the recipe and configure the connection and other resources within it. The integration uses a scheduled orchestration pattern so that you can schedule when to extract the data and how often. The extracted reports are then uploaded to an FTP server. The reports are 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 integration flow performs the following actions:

- Queries bulk data from Oracle ERP Cloud.
- Receives the data via the ERP trigger.
- Invokes it through the SOAP adapter and gets the data from a business event.
- Completes the process. If the export completes successfully as per the Oracle ERP Cloud callback, then uploads the file received in the callback to the FTP server. If the export fails, then sends 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 3*.

System and Access Requirements

- Oracle Integration, Version 23.06.02 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- A secure FTP (sFTP) server or File Server

- An FTP client to access the external sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance and sFTP server in order to successfully connect to these external systems using Oracle Integration and export data in bulk.

Configure Oracle ERP Cloud

To access Oracle ERP Cloud from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud. Additionally, if no report and/or a job for the report is created in Oracle ERP Cloud, you have to create a report and job.

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

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the Integration Specialist role to the user account.
For more information, see Assign Required Roles.
3. Create a report.
 - a. Log in to Oracle ERP Cloud.
 - b. On the Home page, click the arrow next to the user role, and then select **Setup and Maintenance** under **Administration**.
 - c. In the **Search Tasks** field, enter Ledger and click the search icon (magnifying glass) to search for the task.
 - d. In the resulting list, select **Manage Primary Ledgers**.
 - e. Select **Manage Primary Ledgers** again from the task list.
 - f. Note down the **Ledger ID**. You'll need it to configure the report in step 4.
 - g. Now, log in to BI Publisher at <https://<yourinstance>/xmlpserver/servlet/catalog> using a valid user name and password
 - h. In the Catalog window, click the "+" sign on the top-left of the window, and then select **Data Model**.
 - i. In the Data Model window, click the **Diagram** tab, and then click the "+" sign.
 - j. In the resulting menu, select **SQL Query**.
 - k. In the New Data Set - SQL Query window, enter the following details and then click **OK**.
 - i. In the **Name** field, enter a suitable name.
 - ii. In the **Data Source** drop-down field, select **ApplicationDB_FSCM**.
 - iii. In the **Type of SQL** drop-down field, select **Standard SQL**.
 - iv. Add the following SQL query:

```
SELECT
```

```
l.ledger_id, l.name,
```

```

c.code_combination_id, c.account_type, segment1,

segment2, segment3, segment4, segment5, segment6, segment7,
segment8, segment9, segment10,

c.financial_category FROM

gl_ledgers l, gl_code_combinations c

WHERE

c.detail_posting_allowed_flag = 'Y' AND c.summary_flag = 'N'

AND c.enabled_flag = 'Y'

and sysdate between nvl(c.start_date_active, sysdate) and
nvl(c.end_date_active, sysdate)

and c.chart_of_accounts_id = l.chart_of_accounts_id and l.ledger_id
= < ledger_ID >

order by 1

```

Note that ledger ID must be the ledger ID that you noted down in step 3f.

- l. Select the **Data** tab, and then click **View**.
- m. In the Data Model pane, click **Parameters**.
- n. In the Parameters panel, click the + sign to add a parameter. Enter the **Name**, **Data Type**, and **Value** for the parameter. Add as many parameters as you want, and then click **Save**.
- o. Select a folder to save your data. Enter a name and description, and then click **Save**.
You get a confirmation that the data has been saved.
- p. Click **Save as Sample Data**.
- q. Click **OK** in the resulting dialog.
- r. Click **Create Report**.
- s. In the Create Report wizard:
 - i. Select **Use Data Model**. In the **Data Model** field, browse and select the data model that you saved. Click **Next**.
 - ii. Select the **Page Option** and **Layout**, and then click **Next**.
 - iii. Drag and drop the data that you want to include in the report from the left to the right pane. Click **Next**.
 - iv. Select the **Show Grand Totals Row** check box, and click **Preview Report** to view the report. Click **Next**.
 - v. Click **Finish**.
- t. Select a Catalog folder where you want the report to be saved. Enter a name and description and click **Save**.

The report is created. If required, you can change the output format of the report by clicking **View Report** and choosing an available format (for example: PDF, HTML, RTF and so on).

4. Create a job for the report.
 - a. On the Oracle ERP Cloud Home page, click the arrow next to the user role, and select **Setup and Maintenance**.
 - b. Click the **Tasks** icon on the side menu, and then select **Search** from the menu.
 - c. In the search bar, search for scheduler job.
 - d. Select the required **Task List**.
 - e. Select the required task in the task list.
 - f. Click the + sign to create a new job.
 - g. Enter information in the required fields.
 - **Display Name:** Enter a display name.
 - **Name:** Enter a name.
 - **Path:** Enter the folder path to store the job definition.
Note that this folder is created in /oracle/apps/ess/custom/<Job Application Name>. Where, job application name should be FscmEss.
 - **Job Type:** Select **BIPJobType**.
 - **Report ID:** This is the path to the location where your job is saved.
 - h. In the Parameters panel, click the + sign to add a parameter. Enter the **Name**, **Data Type**, and **Value** for the parameter. Add as many parameters as you want and then click **Save**.
Note that the parameters added here must be same as those that were added in step 3n while creating the report.
 - i. Click **Save and Close**.

A message confirms that changes have been saved.

Access Your FTP Server

Perform the following tasks on your FTP server.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an output folder to write files with the bulk data. Note down the path of the directory.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ERP Cloud Connection

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

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

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle SOAP ERP Integration Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the WSDL URL. For example: `https://your_domain_name.fa.DC.oraclecloud.com/publicFinancialCommonErpIntegration/ErpIntegrationService?WSDL`
3. In the Security section, enter the following details:

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle FTP Connection

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

Field	Information to Enter
FTP Server Host Address	Enter the host address of your SFTP server.

Field	Information to Enter
FTP Server Port	22
SFTP Connection	Select Yes from the list.

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

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

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

1. In the Lookups section, click the lookup name.
2. In the **Property_Value** column for **OutputDirectoryOnFTP**, enter the FTP directory path where the BI publisher report has to be uploaded. See [Access Your FTP Server](#).
3. In the **Property_Value** column for **EmailToInCaseOfFault** enter the email where notification emails need to be sent in case of an error.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

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

1. In the project workspace, click **Activate**. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.
A message confirms that the integration has been activated. Refresh the page to view the updated status of the integration.
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.
 - **ParametersForScheduledJob**
 - **ErpScheduledJobName**
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration.
 - a. In the project workspace, click **Observe**. You'll see the integration flow being triggered and running successfully.
 - b. To manage errors in your project, see [Manage Errors in a Project](#).
4. Log in to your FTP server and check if the BI publisher report has been exported to the output folder.

Related Documentation

- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)
- [Using the FTP Adapter with Oracle Integration 3](#)
- [Using File Server in Oracle Integration 3](#)
- [Using the SOAP Adapter with Oracle Integration 3](#)

Export Employee Images from Oracle HCM Cloud to an FTP Server

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

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

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

- The integration flow calls an Oracle HCM custom Enterprise Scheduler (ESS) job using the export bulk data option of the Oracle ERP Cloud adapter.

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

System and Access Requirements

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

Before You Install the Recipe

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

Access Your FTP Server

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

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

Configure Oracle ERP Cloud

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

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

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

Configure Oracle Fusion SaaS

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

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

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

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

Note

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

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle ERP Cloud Connection (Trigger, Invoke)

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

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

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

A message confirms if your test is successful.

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

Configure the Oracle FTP Connection (Trigger, Invoke)

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

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

- In the Security section, enter the following details:

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

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

A message confirms if your test is successful.

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

Configure the Lookup Table

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

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

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

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

Activate and Run the Recipe

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

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

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

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

Related Documentation

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

Export Employee Records from Workday to Azure Active Directory

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the employee records from the Workday instance and exports them to Azure Active Directory.

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

System and Access Requirements

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

- An account on Azure Active Directory with the Administrator role

Before You Install the Recipe

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

Configure Workday

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

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

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

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

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

Configuration details for other related calculated fields are listed below.

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

2. Create a custom report.
 - a. Log in to the Workday instance.
 - b. In the quick find box, search for **Create custom report**.
 - c. In the Create Custom Report window:
 - i. Enter the **Report Name**. For example: RPT_INT_S0A_Test_11.
 - ii. Select the **Report Type** as **Advanced**.
 - iii. Select the **Enable As Web Service** check box.
 - iv. Deselect the **Optimized for Performance** check box.
 - v. Select the **Data Source** as **All Workers**.
 - vi. Click **OK**. You are redirected to Edit Custom Report page.
 - vii. Under the **Columns** tab, add the **Fields** and **Custom Fields** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

Business Object	Field	Column Heading Override	Column Heading Override XML Alias
Worker	First Name	First Name	First_Name
Worker	Last Name		lastName
Worker	Email - Work	Emails	Email_-_Work
Worker	Employee ID		Employee_ID
Worker	Worker Status		Worker_Status
Worker	Date of Birth		dateOfBirth
Worker	Supervisory Organization		Department
Worker	Work Address - City	Primary Address - City	Primary_Address
Worker	Work Address - Country	Primary Address - Country Name	Phone_Work
Worker	Work Address - Formatted Line 1	Primary Address - Formatted Line 1	Company_-_Name
Worker	Work Address - Formatted Line 2	Primary Address - Formatted Line 2	Primary_Address_-_Formatted_Line_2

Business Object	Field	Column Heading Override	Column Heading Override XML Alias
Worker	Phone - Primary Work	Phone - Primary Work	primaryWorkPhone
Worker	Business Title		Business_Title

- d. Click **OK**.
- e. After creating custom reports, you can specify how the report results should be sorted.
- f. Under the **Sort** tab, add the **Fields** and **Sort Direction** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

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

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

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

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

Note

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

3. Create an EIB report.
 - a. Log in to the Workday instance.
 - b. In the quick find box, search for **Create EIB** task.
 - c. Enter the **Name** of the EIB task (RPT_INT_SOA_Test_11).
 - d. In the Create EIB pop-up, choose **Outbound** and click **OK**.
 - e. In the General Settings, enter comments and then click **Next**.
 - f. Under the **Get Data** tab, select the **Data Source Type** as **Custom Report**.
 - g. Select the **Custom Report** as **RPT_INT_SOA_Test_11** and then click **Next**.

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

The EIB report gets generated.

Configure Azure Active Directory

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

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Workday Connection

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

Field	Information to Enter
Host Name	Enter the Workday host name. For example: https://wd2-impl-services1.workday.com.

Field	Information to Enter
Tenant Name	Enter the tenant name of the account created for Oracle Integration on Workday. See Prerequisites for Creating a Connection in <i>Using the Workday Adapter with Oracle Integration 3</i> .

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

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Azure AD Connection

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

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

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

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

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

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

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

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

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

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

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **Oracle Workday Azure AD UserSync** integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update Property Values dialog, update the following:

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

- c. Click **Submit**.

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

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

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

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

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

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

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to Azure Active Directory and check if users are created for employee records exported from Workday.

View the employee records.

- a. In the **Filter** box, search for **Users**.
- b. Under **Users**, select **All Users**.

The list of users created is displayed.

Related Documentation

- [Using the Workday Adapter with Oracle Integration 3](#)
- [Using the Azure Active Directory Adapter with Oracle Integration 3](#)

Export Employee Records from Workday to SAP

Use this recipe to export employee records from Workday to SAP.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe exports Workday employee records to SAP, according to a schedule specified in Oracle Integration.

To use this recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the employee records from the Workday instance and exports them to SAP.

While running the recipe, you can configure the integration flow to read Workday employee records created after a specified time, thereby picking up only the new or required employee records in each run. Basic data such as first name, last name, employee type, nationality, hire date, and so on are synchronized between Workday and SAP.

System and Access Requirements

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

Before You Install the Recipe

You must perform the following configuration tasks on your Workday and SAP instances in order to successfully connect to these external systems using Oracle Integration and export employee records from Workday to SAP.

Configure Workday

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

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

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

1. Create calculated fields.
 - a. In the quick find box, search for **Create Calculated Fields** task.
 - b. In the pop-up window, enter the **Field name**. For example: CF_hired_updated. Also, select the **Business Object** as **Worker**, and **Function** as **Evaluate Expression**.
 - c. Click **OK**. You are redirected to **Create Calculated Field - Evaluate Expression** page.
 - d. Under Calculation, select the **Field Type** as **Text** and **Default Value** as **Empty String**.
 - e. In the Condition column, create the following calculated fields:
 - **CF_SAP_is_updated?**

And/Or	Field	Calculated Field Details	Operator	Comparison Type	Comparison Value
And	CF_Legal_name_change1	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_legal_change1. See step g . Sort Field: Effective Date Sort Direction: Descending (Z to A) Instance to be Returned: First Occurrence Function: Extract Single Instance	is not empty	NA	NA
Or	CF_new_events	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_effective date range. See step g . Fields To Aggregate: Business Process Event Function: True or False	is not empty	NA	NA

- **CF_SAP_hired?**

And/Or	Field	Calculated Fields Details	Operator	Comparison Type	Comparison Value
And	Hire Date	NA	greater than or equal to	Value from another field	Prompt-Start Date
And	CF_legal_change1	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_legal_change1. See step g . Sort Field: Effective Date Sort Direction: Descending (Z to A) Instance to be Returned: First Occurrence Function: True or False	is empty	NA	NA

And/Or	Field	Calculated Fields Details	Operator	Comparison Type	Comparison Value
And	CF_new_events	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_effective change. See step g . Fields To Aggregate: Business Process Event Function: True or False	is empty	NA	NA

- f. In the **Return Value If Condition is True** column, create the following calculated fields.

Calculated Field	Details
CF_sap_updated	Business Object: Worker Text Constant: UPDATED
CF_sap_hired	Business Object: Worker Text Constant: HIRED

- g. Also, configure the following calculated fields with the following details.

- CF_SAP_Integration_Status

Calculated Field	Details
CF_SAP_Integration_Status	Field Name: CF_SAP_Integration_Status Business Object: Worker Calculation: Lookup Related Value: Lookup Field: Worker Additional Data Related Business Object: Worker Custom Data Snapshot Return Value: SAP_Integration_Status

- CF_legal_change1

And/Or	Field	Operator	Comparison Type	Comparison Value
And	Business Process Type	in the selection list	Value specified in this filter	Legal Name Change
And	Status	greater than or equal to	Value specified in this filter	Successfully Completed
And	Effective Date	greater than or equal to	Value from another field	Prompt - Start Date

- CF_effective change

And/Or	Field	Calculated Field Details	Operator	Comparison Type	Comparison Value
And	Business Processes Definition	Business Object: Action Event Function: True or False	in the selection list	Value specified in this filter	— Change Job (Default Definition) — Change Job for Global Modern Services — Change Job for IT Services Group — Change Job for Sales — Change Organization Assignments for Worker (Default Definition) — Change Organization Assignments for Worker for Field Sales - North America Group — Change Organization Assignments for Worker for Global Modern Services — Change Organization Assignments for Worker for Marketing — Change Organization Assignments for Workers by Organization (Default Definition) — Change Organization Assignments for Workers by Organization for Global Modern Services — Contact Change (Default Definition) — Move Worker (Supervisory) (Default Definition) — Move Worker (Supervisory) for Global Modern Services — Move Workers (By Organization) (Default Definition) — Move Workers (By Organization) for Global Modern Services — Move Workers (Supervisory) (Default Definition) — Move Workers (Supervisory) for

And/Or	Field	Calculated Field Details	Operator	Comparison Type	Comparison Value
					Global Modern Services
					– Personal Information Change (Default Definition)
					– Personal Information Change for Global Modern Services
					– Switch Primary Job (Default Definition)
					– Switch Primary Job for Global Modern Services
					– Title Change (Default Definition)
					– Title Change for Global Modern Services
					– Work Contact Change (Default Definition)
And	Status	Business Object: Action Event Function: True or False	equal to	Value specified in this filter	Successfully Completed
And	Effective Date	Business Object: Action Event Function: True or False	greater than or equal to	Value from another field	Prompt - Start Date

Note

Business process names mentioned in the tables above are subject to change based on the Workday instance configuration. You need to select business processes according to your configuration.

2. Create a custom report.
 - a. Log in to the Workday instance.
 - b. In the quick find box, search for **Create custom report**.
 - c. In the Create Custom Report window:
 - i. Enter the **Report Name**. For example: Workday_To_SAP_ERP_Worker_Report.
 - ii. Select the **Report Type** as **Advanced**.
 - iii. Check the **Enable As Web Service** box.
 - iv. Uncheck the **Optimized for Performance** box.
 - v. Select the **Data Source** as **Active and Terminated Workers**.
 - vi. Click **OK**. You are redirected to Edit Custom Report page.

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

Business Object	Fields	CColumn Heading Override oXML Alias
		I u m n H e a d i n g O v e r r i d e
Worker	Employee ID	WWorker_ID o r k e r _I D
Worker	Worker Type	WWorker_Type o r k e r _T y p e
Worker	Employee Type	EEmployee_Type m p l o y e e _T y p e

Business Object	Fields	CColumn Heading Override oXML Alias
		l u m n H e a d i n g O v e r r i d e
Worker	Contingent Worker Type	CContingent_Worker_SubType oe n t i n g e n t - W o r k e r - T y p e
Worker	Hire Date	HHire_Date i r e - D a t e
Worker	Termination Date - All	EEnd_Date n d - D a t e

Business Object	Fields	CColumn Heading Override oXML Alias
		l u m n H e a d i n g O v e r r i d e
Worker	First Name	FFirst_Name i r s t - N a m e
Worker	Last Name	LLast_Name a s t - N a m e
Worker	Gender	GGender e n d e r
Worker	Date of Birth	BBirth_Date i r t h - D a t e

Business Object	Fields	CColumn Heading Override oXML Alias
		l u m n H e a d i n g O v e r r i d e
Worker	Primary Nationality (Locale Sensitive)	NNationality a t t i o n a l i t y
Display Language	User Language Code	LLanguage a n g u a g e
Worker	Country of Birth	CCountry_of_Birth o u n t r y _ o f _ B i r t h

Business Object	Fields	CColumn Heading Override oXML Alias
		l u m n H e a d i n g O v e r r i d e
Location Address - Country	Alpha-2 Code	CCountry o u n t r y
Worker	Job Profile	JJob_Profile o b P r o f i l e
Company	Reference ID	CCompany_Org_ID o m p a n y O r g a n i z a t i o n I D

Business Object	Fields	Column Heading Override oXML Alias
		l u m n H e a d i n g O v e r r i d e
Region	Reference ID	RRegion_Org_ID e g i o n - O r g - I D
Location	Reference ID	LLocation_Org_ID o c c a t i o n - O r g - I D

Business Object	Fields	Column Heading Override XML Alias
		Column Heading Override
Supervisory Organization	Reference ID	Supervisory_Org_ID
Cost Center	Reference ID	CostCenter_Org_ID

Business Object	Fields	Column Heading Override XML Alias
		Position_Heading_Override
Worker	Position Time Type	Position_Time_Type Position Time Type
Worker	CF_hired_updated	Action Condition
Worker	Position ID	Position_ID Position ID

Business Object	Fields	Column Heading Override XML Alias
Worker	Pay Group	PPay_Group a y _G r o u p
Worker Additional Data	SAP_Integration_Status	NSAP_Integration_Status A
Worker Additional Data	SAP_Personnel_Number	NSAP_Personnel_Number A

- d. Click **OK**.
- e. After creating custom reports, on the **Sort** tab, you can specify how the report results should be sorted.
- f. On the **Filter** tab, add the fields required for the employee details along with the calculated fields and the relevant business object.

And/Or	Fields	Operator	Comparison Type	Comparison Value
And	Terminated (based on report date)	equal to	Value specified in this filter	NA
And	Hire Date	greater than or equal to	Prompt the user for the value	Starting Prompt
Or	CF_Legal_name_change1	is not empty	NA	NA
Or	CF_new_events	is not empty	NA	NA
Or	CF_SAP_Integration_Status	equal to	Value specified in this filter	CreateFailed
Or	CF_SAP_Integration_Status	equal to	Value specified in this filter	UpdateFailed

- g. In the Prompts section, check the **Populate Undefined Prompt Defaults** box.

Note

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

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

The EIB report is generated.

Configure SAP

To access SAP using Oracle Integration and import employees, you must perform certain configuration tasks on your SAP instance.

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

In addition, you must perform the following configurations that are specific to this recipe.

1. Configure SAP Inbound parameters for the HRMD_A message type using WE20. See SAP Inbound Communication in *Using the SAP Adapter with Oracle Integration 3*.
2. Create a function module using SE37.

Note

Before creating a function module, ensure that you meet the following prerequisites:

- You have access to an SAP system based on either SAP Netweaver or ABAP platform.
- You have installed the SAP GUI on your local system.
- You have authorization to access SE37.
- You have the developer key for the SAP system based on Netweaver.
- You have created the ABAP Function Group.

You must create a Function Group **ZFG1**, save it in a package **ZNUMBER**, and activate the Function Group.

- a. After you have met the prerequisites, go to the transaction SE37 (ABAP Function Modules). Enter SE37 in the command field and press Enter.
- b. In the **Function Module** field, enter the name of the function module (for example, NUMBER_GET_NEXT), then click **Create**.

Note

Most SAP ABAP artifacts need to start with a “Z” or “Y”.

- c. In the Copy Function Module window, copy the Function Module name, NUMBER_GET_NEXT to ZNUMBER_GET_NEXT, and enter the Function Group name as ZFG1, then click **Copy**.
- d. Activate the Function Module (**ZNUMBER_GET_NEXT**) and click **Test**.
- e. In the initial screen of the Test Function Module, enter the following values and click **Execute**.

Import Parameters	Value
NR_RANGE_NR	02
OBJECT	RP_PREL
QUANTITY	00000000000000000001
SUBOBJECT	none
TOYEAR	none
IGNORE_BUFFER	none

- f. In the result screen of the Test Function Module, select **Number Details**, and get the next NUMBER.

Install and Configure the Recipe

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Oracle Workday Connection

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

Field	Information to Enter
Host Name	Enter the Workday host name. For example: <code>https://wd2-impl-services1.workday.com</code>
Tenant Name	Enter the tenant's name of the account created for Oracle Integration on Workday. See Configure Workday .

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

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

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

A message confirms if your test is successful.

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

Configure the Oracle SAP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, upload the adapter inbound properties file. See JCO Connection Properties Files in *Using the SAP Adapter with Oracle Integration 3*.
3. In the Security section, enter the following details:

Field	Information to Enter
Username	Enter the username to connect to the SAP instance. See Configure SAP .
Password	Enter the password to connect to the SAP instance.

4. In the Access type section, select an agent group for accessing the service hosted on your premises behind the firewall.

- a. Select **Connectivity agent** and click **Associate agent group**.
- b. In the resulting window, select an agent group and click **Use**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

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

Configure the Lookup Tables

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

- **WORKDAY_SAP_COUNTRY_CODE**: Use this lookup to map Country Codes between Workday and SAP.
- **WORKDAY_SAP_PERSONNEL_SUBAREA**: Use this lookup to map Location Org Code in Workday to Personnel Subarea in SAP.
- **WORKDAY_SAP_PERSONNEL_AREA**: Use this lookup to map Region Org Code in Workday to Personnel Area in SAP.
- **WORKDAY_SAP_EMPLOYEE_SUBGROUP**: Use this lookup to map Worker Subtype in Workday to Employee Subgroup in SAP.
- **WORKDAY_SAP_GENDER**: Use this lookup to map Genders between Workday and SAP.
- **WORKDAY_SAP_LANGUAGE**: Use this lookup to map Languages between Workday and SAP.
- **WORKDAY_SAP_EMPLOYEE_GROUP**: Use this lookup to map Worker Type in Workday to Employee Group in SAP.
- **WORKDAY_SAP_EMPLOYEE_PROPERTIES**: Use this lookup to store the properties configured for this recipe as shown in the following table.

Property	Description	Value
WorkdayReportName	Name of the Workday EIB report.	EIB_Workday_To_SAP_ERP_Worker_Report
Client	SAP Client Id	800
SenderPort	SAP port created for Oracle Integration	A000000072
SenderPartnerNumber	SAP partner profile created for Oracle Integration	OIC_SAP_WD
PlanVersion	SAP plan version	1
ObjectType	SAP object type for employee	P
AssignmentObjectType	SAP assignment object type	S
NumberRangeNumber	SAP number range number for employee personnel number (PERNR)	2
NumberRangeObject	SAP number range object for employee personnel number (PERNR)	RP_PREL
IdocStatusDelimiter	Delimiter for SAP EDIDS table query result	
EndDate	SAP default end date	99991231
HiringActionType	SAP action type for employee hiring action	1

Property	Description	Value
HiringActionReason	SAP reason for action for employee hiring action	6
CompanyCode	SAP company code	1000
OrganizationalKey	SAP organizational key	1000
OrganizationalUnit	SAP organizational unit	1000
AdministratorGroup	SAP administrator group	1000
ControllingArea	SAP controlling area	1000
NameFormatIndicator	SAP name format indicator	0
FormOfAddress	SAP form of address	1

1. In the Configuration Editor, click **Lookups** to view the lookup tables in the recipe. Click the name of the lookup table to edit.
2. In the **Value** column, enter or map the appropriate values for the properties.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the Configuration Editor, click **Go back** .

Activate and Run the Recipe

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

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the following property values for the integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
workdayWaitMinutes	Time to wait in minutes for the Workday EIB report to complete.
dvmDefaultValue	Default value for DVM lookups.
emailSubjectFailure	Subject of the email notification sent whenever importing the employee records from Workday to SAP is unsuccessful.
workdayTimeZoneDuration	Difference in time zone between Oracle Integration and Workday in dayTimeDuration format. For example: -PT8H or PT8H.
emailFrom	Sender email address from which notifications are sent.
retryLimit	Number of retries when an invoke fails.
emailSubjectSuccess	Subject of the email notification sent whenever importing the employee records from Workday to SAP is successful.

- c. Click **Submit**.

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

3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The recipe now exports employee records from Workday to SAP.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to SAP and check for the employee records created.
 - a. Enter PA20 in the transaction command window and press Enter.
 - b. Enter the **Personal Number** of the required employee record, and press Enter.

Related Documentation

- *Using the SAP Adapter with Oracle Integration 3*
- *Using the Workday Adapter with Oracle Integration 3*

Export Google Gmail Attachments to an FTP Server

Use this recipe to export email attachments from Google Gmail to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 and configure the connections and other resources within the recipe. 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

Before You Install the Recipe

You must perform the following configuration tasks on your Gmail instance and sFTP server in order to successfully connect to these external systems using Oracle Integration and export email attachments from Google Gmail to an FTP server.

Configure Gmail

You must obtain the Gmail API credentials, that is the Client ID and Secret, for connecting to Gmail from Oracle Integration.

1. Log in to your Google account and access the Google Developer console at <https://console.developers.google.com>.
2. 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 drop-down list after saving the new project. You'll be able to view the project's dashboard only if you select the project.
4. 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.

Note

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 3*.

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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

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

Configure the Google Mail Connection

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

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Gmail. See Configure Gmail .
Client Secret	Enter the client secret obtained while configuring Gmail.
Scope	Enter <code>https://www.googleapis.com/auth/gmail.modify</code> .

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
 - a. 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.
- c. Follow the prompts till you see the access allowed page.
Switch back to your Google Mail Connection window in Oracle Integration.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the FTP Read Write Connection

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

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

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

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

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Update Integration Properties

You must update the integration property values so that the attachments from Gmail are exported to the right folder in the FTP server.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, note the property that has been configured: `Ftpfolderdestination`.
3. 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](#).

4. Click **Submit**.

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

Activate and Run the Recipe

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

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

You've now successfully submitted the integration for execution.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to the FTP server and check if the attachments from Gmail have been exported to the specified directory.

Related Documentation

- [Using the Google Gmail Adapter with Oracle Integration 3](#)
- [Using the FTP Adapter with Oracle Integration 3](#)

Export Invoices from PayPal to an FTP Server

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

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

Before You Install the Recipe

You must perform the following configuration tasks on your PayPal and FTP instances in order to successfully connect to these external systems using Oracle Integration and export invoices.

Configure PayPal

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

1. 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.

- a. Log in to the PayPal Developer Dashboard (<https://developer.paypal.com/developer/applications>) with your account username and password.
- b. In the left menu, under **DASHBOARD**, click **My Apps & Credentials**.
- c. 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.

- d. Note the client ID and client secret. You'll use these later while configuring the PayPal connection in Oracle Integration.

2. Create invoices in your PayPal Sandbox account.

a. Get your Sandbox business account credentials.

- i. Log in to the PayPal Developer Dashboard using your account user name and password.
- ii. In the left menu, under **SANDBOX**, click **Accounts**. The default (system generated) personal and business accounts are listed.
- iii. Select the account of type **Business**, click  in the **Manage accounts** column, and then select **View/edit account**.
- iv. In the resulting Account details dialog, note the **Email ID** and the **System Generated Password**.

- b. Log in to your PayPal Sandbox account using the business account credentials (Email ID and Password) obtained in step 2a.

- c. 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**.

- d. Click **Create Invoice**.

- e. In the Create invoice page, configure the invoice fields.

- i. In the **Bill to** field, enter the customer's name. For example, John Doe.
- ii. Click the **Add** link next to **Ship to**. In the resulting dialog, enter the customer's shipping and billing address. Click **Save**.

- iii. Add invoice items. Enter the item name, quantity, price, and tax.
- iv. Click **Add another line item** to add more items in the invoice.
- f. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the PayPal Connection

1. In the Connections section, click the connection name.
2. In the Properties section, from the **PayPal Environment** drop-down list, select the environment (**Sandbox**).
3. In the Security section, from the **Security Policy** drop-down list, select **Client Credentials**.
4. In the **Client Id** and **Client Secret** fields, enter the Client ID and Secret that you obtained before. See [Configure PayPal](#).
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Configure the FTP Read Write Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the REST PayPal Connection

- In the Connections section, click the connection name.
- From the **Connection Type** drop-down list, select **REST API Base URL**.
- 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**.
- In the **Username** field enter the Client ID, and in the **Password** field enter the Client Secret that you obtained before.
See [Configure PayPal](#).
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure Lookups

Configure lookups to associate values used by one application (PayPal) to values used by the other application (FTP).

- In the Lookups section, click the lookup name.
- 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 <code>/oracle-int/paypal-invoices</code> .
invPrefix	Prefix of the invoice files. For example, if the file name is <code>detail_INV-NOV2020.JSON</code> then the prefix is detail_ .
pageSize	Number of files that are to be fetched per call from PayPal. For example 1 .

- Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, 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.
 - c. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to 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 3](#)
- [Using the FTP Adapter with Oracle Integration 3](#)

Export Items in Bulk from Oracle ERP Cloud to Shopify

Use this recipe to export items in bulk from Oracle ERP Cloud to Shopify.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates products in Shopify whenever items are published in Oracle ERP Cloud (Product Hub).

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When items in Oracle ERP Cloud are published for Shopify (spoke

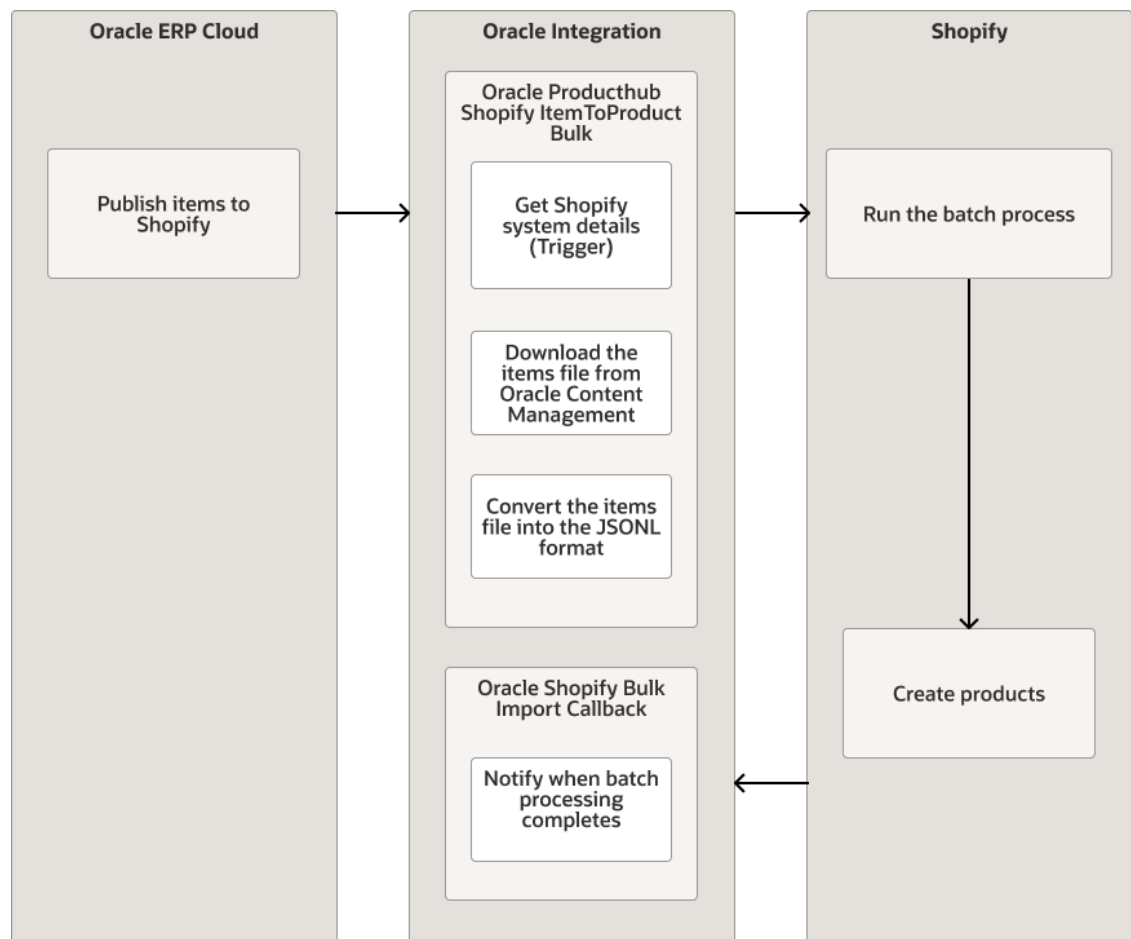
system), the recipe is triggered, and it creates corresponding products in bulk in Shopify. If any records fail, they are stored in a folder in FTP.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- Shopify
- An account on Shopify with the Administrator role
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Recipe Schema

This section provides an architectural overview of the recipe.



This recipe has the following integration flows:

- **Oracle Producthub Shopify ItemToProduct Bulk:** This integration flow is triggered when items in Oracle ERP Cloud are published for Shopify.

The items are then uploaded to a specific location in Oracle Content Management. When triggered, this integration performs the following in Oracle Integration:

- Validates if the published items are for Shopify.
- Verifies the Items file.
- Downloads the Items file from Oracle Content Management.
- Converts the Items file to JSONL format.
- Uploads the JSONL file to Shopify

The `bulkOperationRunMutation` batch process is triggered (from Oracle Integration) and imports data into Shopify. If the batch process throws an exception, a JSONL file with error data is saved in the `newFile.txt` file in the following path in FTP: `/ShopifyERPError_jsonl/newFile.txt`.

- **Oracle Shopify Bulk Import Callback:** This integration flow is triggered when items are successfully created in Shopify. When triggered, this integration flow sends a notification email to the configured email IDs.

Assumptions

- The Oracle ERP Cloud instance has items to be published for Shopify. See [Define Items and Item Structures](#).
- The Oracle ERP Cloud instance has Spoke system for Shopify. See [Product Spoke Systems](#). Ensure that the name of the Spoke system contains Shopify (case-sensitive).

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Shopify instances in order to successfully connect to these external systems using Oracle Integration and export items in bulk.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Shopify

To access the Shopify instance from Oracle Integration, you must perform certain configurations on your Shopify instance. See *Prerequisites for Creating a Connection in Using the Shopify Adapter with Oracle Integration 3*.

Also, register the Oracle Shopify Bulk Import Callback integration flow for Shopify webhook calls.

1. Send a POST request to the following REST endpoint: `https://shopify_Store_hostname>/admin/api/2022-04/graphql.json`. Specify the payload mentioned on the cURL tab in [Create an HTTP Webhook in Shopify](#).
2. In the payload, update the following fields with the necessary data.

Field	Information to Enter
Request Headers (-H)	Enter 'X-Shopify-Access-Token=<shopifyShared Secret>'.
topic	Enter BULK_OPERATIONS_FINISH.
callbackurl	Enter <code>https://<username:password>@<OIC_hostname>/ic/api/integration/v2/flows/rest/project/ORCL-R-PDH_SHOPIFY_ITEMVAR_SYNC/ORCL-R-SHOPIFY_BULK_IMPORT_CALBK/1.0/onbulkimportfinish</code> .

3. Validate the registered webhook. Send a GET request to the following REST endpoint:

`https://<shopify_Store_hostname>/admin/api/2023-04/webhooks.json`
Request Headers- X-Shopify-Access-Token=<shopifyShared Secret>

Access Your FTP Server

Obtain an FTP server and ensure that you can access it.

1. Login to the server using your username and password through an FTP client; for example, FileZilla
2. Create a directory with the name `ShopifyERPError_jsonl` on the server to store the error response.

Note down the path of the directory.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Shopify REST Connection

- In the Connections section, click the connection name.
- In the Properties section, in the **Hostname** field, enter the URL of your Shopify instance. For example: `https://your_domain_name.myshopify.com/`
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the API key obtained earlier. See Configure Shopify .
Password	Enter the Shared Secret obtained earlier.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

- In the Connections section, click the connection name.
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the FTP Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

- a. **FTP Server Host Address:** Enter the host address of your sFTP server.
 - b. **FTP Server Port:** Enter 22.
 - c. **SFTP Connection:** specify the value as Yes.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your sFTP server.
Private Key File	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the UCM Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the WSDL URL. For example: `https://erp-hostname/idcws/GenericSoapPort?wsdl`
3. In the Security section, enter the following details:

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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe. Publish items for Shopify.
 - a. From the menu, select **Manage Spoke System**.
 - b. In the Edit Spoke System page, select the publish criteria if required.
 - c. Click **Publish**. From the drop-down list, select **Publish Now**.

- d. In the Publish Now pop-up, specify the **Publish from date**, and click **Submit**.
- e. Click **OK**.

You've now published items and triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to Shopify and check for the new products. See [Searching for products and filtering your product list](#).

Related Documentation

- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Export Items with Variants from Oracle ERP Cloud to Shopify

Use this recipe to export items along with variants from Oracle ERP Cloud to Shopify.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates products in Shopify whenever items are published in Oracle ERP Cloud (Product Hub).

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When items in Oracle ERP Cloud are published for Shopify (spoke system), the recipe is triggered, and it creates corresponding products in bulk in Shopify.

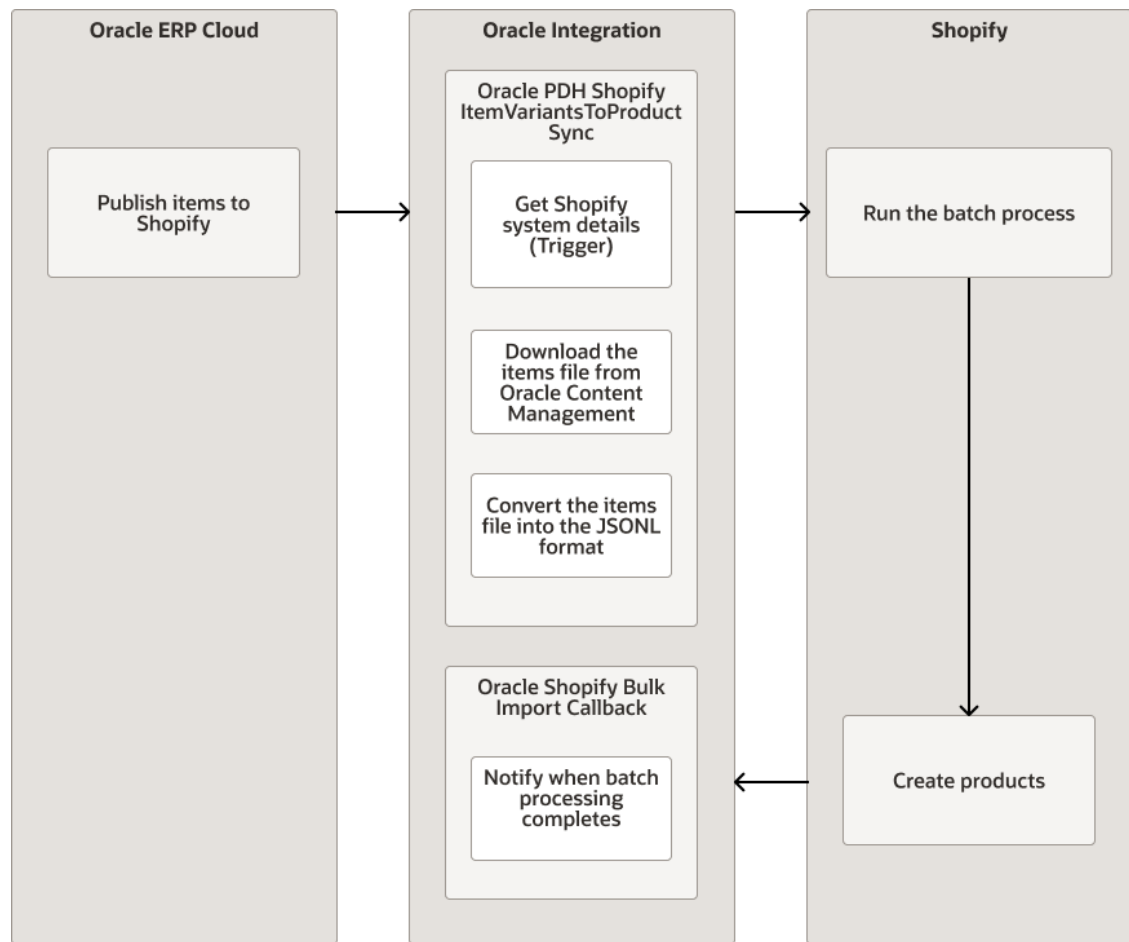
This recipe exports item variants along with items from Oracle ERP Cloud to Shopify. If any records fail, they are stored in a folder in FTP.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- Shopify
- An account on Shopify with the Administrator role
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Recipe Schema

This section provides an architectural overview of the recipe.



This recipe has the following integration flows:

- Oracle PDH Shopify ItemVariantsToProduct Sync:** This integration flow is triggered when items in Oracle ERP Cloud are published for Shopify. The items are then uploaded to a specific location in Oracle Content Management. When triggered, this integration performs the following in Oracle Integration:
 - Validates if the published items are for Shopify.
 - Verifies the Items file.
 - Downloads the Items file from Oracle Content Management.
 - Converts the Items file to JSONL format.
 - Uploads the JSONL file to Shopify.

The `bulkOperationRunMutation` batch process is triggered (from Oracle Integration) and imports data into Shopify. If the batch process throws an exception, a JSONL file with error data is saved in the `newFile.txt` file in the following path in FTP: `/ShopifyERPError_jsonl/newFile.txt`.

- Oracle Shopify Bulk Import Callback:** This integration flow is triggered when items are successfully created in Shopify. When triggered, this integration flow sends a notification email to the configured email IDs.

Assumptions

- The Oracle ERP Cloud instance has items to be published for Shopify. See [Define Items and Item Structures](#).
- The Oracle ERP Cloud instance has Spoke system for Shopify. See [Product Spoke Systems](#). Ensure that the name of the Spoke system contains Shopify (case-sensitive).

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Shopify instances in order to successfully connect to these external systems using Oracle Integration and export items in bulk.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Shopify

To access the Shopify instance from Oracle Integration, you must perform certain configurations on your Shopify instance. See Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Also, register the Oracle Shopify Bulk Import Callback integration flow for Shopify webhook calls.

1. Send a POST request to the following REST endpoint: `https://shopify_Store_hostname>/admin/api/2022-04/graphql.json`. Specify the payload mentioned on the cURL tab in [Create an HTTP Webhook in Shopify](#).
2. In the payload, update the following fields with the necessary data.

Field	Information to Enter
Request Headers (-H)	Enter 'X-Shopify-Access-Token=<shopifyShared Secret>'.
topic	Enter BULK_OPERATIONS_FINISH.

Field	Information to Enter
callbackurl	Enter https:// <username:password>@<OIC_hostname>/ic/api/ integration/v2/flows/rest/project/ORCL-R- PDH_SHOIFY_ITEMVAR_SYNC/ORCL-R- SHOIFY_BULK_IMPORT_CALBK/1.0/ onbulkimportfinish.

3. Validate the registered webhook. Send a GET request to the following REST endpoint:

https://<shopify_Store_hostname>/admin/api/2023-04/webhooks.json
Request Headers- X-Shopify-Access-Token=<shopifyShared Secret>

Access Your FTP Server

Obtain an FTP server and ensure that you can access it.

1. Login to the server using your username and password through an FTP client; for example, FileZilla
2. Create a directory with the name ShopifyERPErrors on the server to store the error response.

Note down the path of the directory.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: https://your_domain_name.fa.DC.oraclecloud.com.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Shopify REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, in the **Hostname** field, enter the URL of your Shopify instance. For example: `https://your_domain_name.myshopify.com/`
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the API key obtained earlier. See Configure Shopify .
Password	Enter the Shared Secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Save**. If prompted, click **Save** again.
3. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

4. To return to the project workspace, click **Go back** .

Configure the FTP Connection

Use the following steps to configure the **Oracle Connection_ERP_jsonL_Shopify Connection**.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:
 - a. **FTP Server Host Address**: Enter the host address of your SFTP server.
 - b. **FTP Server Port**: Enter 22.
 - c. **SFTP Connection**: specify the value as Yes.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your SFTP server.

Field	Information to Enter
Private Key File	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the UCM Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the WSDL URL. For example: `https://erp-hostname/idcws/GenericSoapPort?wsdl`
3. In the Security section, enter the following details:

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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

1. In the Lookups section, click the lookup name.
2. Enter the field names and corresponding values. Refer to the details provided for each table below.

The recipe contains the following three lookup tables. Edit them as necessary.

- **ORACLE-BRT-PDH_SHOPIFY_CONSTANTS:** Used to map WeightUnit of items in Shopify with the corresponding WeightUnit of items in Oracle ERP Cloud.

Weight Unit in Oracle ERP Cloud	Weight Unit in Shopify
kg	KILOGRAMS

- **ORACLE-BRT-ERP_Shopify_PdSync_Notification_Const:** Used to configure the email address to which the error notifications are sent when the recipe fails.

Email_Property	Email_Value
fromAddress	Specify the email address from which the notification emails are sent. Enter an email address of your choice.
toAddress	Specify the email address to which the notification emails are sent. Enter an email address of your choice.

- **ORACLE-BRT-PDH_SHOPIFY_CONSTANTS_VARIANTS:** Used to configure item variants based on item category.

Item_Category _Code	ItemEffCatego ry	Variant_Opti ons	Variant_Opti on1	Variant_Opti on2	Variant_Opti on3
Smartphone	ItemEFFBsm_5 Fvariants_5Fat gPrivateVO	["Language"," Color","Vtag e"]	_Language	_Color	_Voltage
Smartwatch	ItemEFFBsw_5 Fvariants_5Fat gPrivateVO	["Material","C olor"]	_Material	_Color	

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe. Publish items for Shopify.
 - a. From the menu, select **Manage Spoke System**.
 - b. In the Edit Spoke System page, select the publish criteria if required.
 - c. Click **Publish**. From the drop-down list, select **Publish Now**.
 - d. In the Publish Now pop-up, specify the **Publish from date**, and click **Submit**.
 - e. Click **OK**.

You've now published items and triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to Shopify and check for the new products. See [Searching for products and filtering your product list](#).

Related Documentation

- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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.

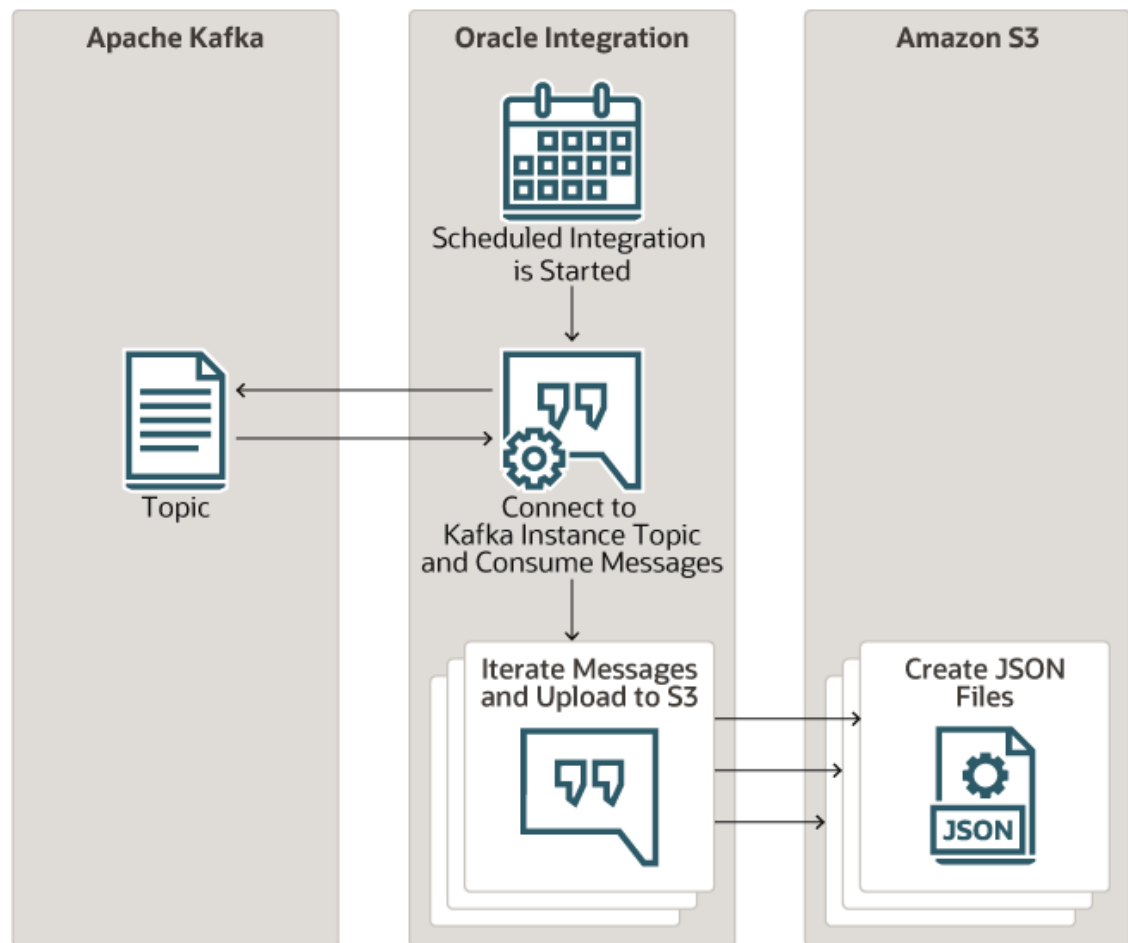
To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow manually or specify an execution schedule for it. When triggered, the integration flow queries for messages in the specified 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.

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.

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**.
5. Get the Apache Kafka Cluster Bootstrap Server URL.
 - a. In the left navigation pane, expand **Cluster overview** and then click **Cluster settings**.
 - b. 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 in *Using the Apache Kafka Adapter with Oracle Integration 3*.

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.

1. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Apache Kafka Connection

1. In the Connections section, click the connection name.
2. In the **Bootstrap Servers** field, enter the Bootstrap Server URL you obtained earlier while configuring Apache Kafka. See [Configure Apache Kafka](#).
3. In the **Security Policy** field, select **SASL PLAIN Over SSL**.
4. 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](#).
5. 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 password in the **KeyStore password** field.See [Configure Apache Kafka](#).
6. Click **Save**. If prompted, click **Save** again.

- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the REST Amazon S3 Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter <code>https://s3.<your-bucket-region>.amazonaws.com/</code> . Where your <code>-bucket-region</code> is the region in which you had set up your Amazon S3 bucket. For example, <code>https://s3.eu-central-1.amazonaws.com/</code> . See Configure Amazon S3 .

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select AWS Signature Version 4 .
Access Key	Enter the access key that you obtained. See Configure Amazon S3 .
Secret Key	Enter the secret key that you obtained.
AWS Region	Select the AWS region where you have created the Amazon S3 bucket. For example, <code>eu-central-1</code> .
Service Name	Select Amazon Simple Storage Service (Amazon S3) .

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Integration Flow

Configure the Apache Kafka adapter endpoint properties in the integration.

- In the Integrations section, click the integration flow name.
- On the integration canvas, click the **LoadTopicMessages** element, and then click **Actions**  and select **Edit**.

The Apache Kafka Endpoint Configuration Wizard with its Basic Info page is displayed.

- Click **Next** in the Basic Info page.
- In the Operation page that appears, click **Next** again.

The Topic & Partition page is displayed.

5. Configure the following in the Topic & Partition page.
 - a. In the **Select a Topic** list, select 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.
 - c. 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 3*.
6. Click **Save**. If prompted, click **Save** again.
7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

Note

Before activating and running the recipe, ensure that there are some messages in the Apache Kafka topic.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can 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.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

4. Check if the Apache Kafka messages are exported into the Amazon S3 bucket as JSON files.
 - a. 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 3*
- *Using the REST Adapter with Oracle Integration 3*

Export New Employee Data from Oracle HCM Cloud to an FTP Server

Use this recipe to export new hire information from Oracle HCM Cloud to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe uses the ATOM feed in the Oracle HCM Cloud Adapter to subscribe to new hire notifications in Oracle HCM Cloud, 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.

To use this recipe, you must install the recipe and configure the connections and other resources within it. In the first run, you activate and run the integration flow of the recipe by submitting an initial date time value for the schedule parameter **ATOMLastRunDateTime**. In subsequent runs, this value will update to reflect the actual date and time values. The recipe fetches new employee data from Oracle HCM Cloud in a JSON (`getEmployeeResponse.json`), formats the data in a comma separated value (CSV) file (`newEmployeeFile.csv`), and writes the CSV file to an FTP server. You can download the sample files to use in the recipe from [Sample files](#).

System and Access Requirements

- Oracle Integration, Version 23.06.02 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- A secure FTP (SFTP) server

- An FTP client to access the external sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud instance and sFTP server in order to successfully connect to these external systems using Oracle Integration and export new employee details.

1. Complete the prerequisite tasks in your Oracle HCM Cloud instance. See *Prerequisites for Creating a Connection* in *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.
2. Access an FTP server and create the required output folder.
 - a. Log in to the server using your username and password through an FTP client; for example, FileZilla.
 - b. Create a output folder on the server to store the CSV files with the new employee data. Note down the path of the folder.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on the Oracle HCM Cloud instance. See <i>Prerequisites for Creating a Connection</i> in <i>Using the Oracle HCM Cloud Adapter with Oracle Integration 3</i> .
Password	Enter the password of the account created for Oracle Integration on the Oracle HCM Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST HCM Connection

1. In the Connections section, click the connection name.

- In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the HCM REST URL in the format: <code>https://<host>:<port>/hcmRestApi/resources/latest</code> .

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of the account created for Oracle Integration on the Oracle HCM Cloud instance. See <i>Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3</i> .
Password	Enter the password of the account created for Oracle Integration on the Oracle HCM Cloud instance.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the FTP Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

- 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.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the following integration property with appropriate value.
 - **OutputFolder**: Enter the folder path where the output files need to be placed. See [Before You Install the Recipe](#).
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the schedule parameter **ATOMLastRunDateTime**. For example:
2018-01-01T00:00:00.000Z
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to the FTP server and check if a CSV file with the new employee data has been uploaded to the specified input folder.

Related Documentation

- [Using the Oracle HCM Cloud Adapter with Oracle Integration 3](#)
- [Using the FTP Adapter with Oracle Integration 3](#)

Export New Hire Information from Oracle HCM Cloud

Use this recipe to extract new hire information from Oracle HCM Cloud and export the file to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe extracts new hire information from Oracle HCM Cloud and then uses FTP to export the file to an FTP server, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the new hire information from Oracle HCM Cloud instance and exports the file to an FTP server.

System and Access Requirements

- Oracle Integration
- Oracle HCM Cloud, Release 13 or later
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud instance and sFTP server in order to successfully connect to these external systems using Oracle Integration and export new hire information.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and export new hire information, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Also, configure the HCM Extract that you load into your Oracle HCM Cloud instance. See [New Hire Export](#). The **Active_Employee_Feed.xml** file contains the HCM Extract definition.

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 on the server into which the exported files will be written. Note down the path of the directory.

Also, ensure that you have write permissions on the sFTP server directory into which you want to export the files.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the FTP Read Write Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your SFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your SFTP server.
Password	Enter the password to connect to your SFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

- a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
- b. On the Configure and run page, specify a value for the `Schedule_parameter_DocumentID` parameter.

This parameter stores the last successfully-retrieved document identifier. Enter 0 in the **New Value** field to fetch all the new hire records from Oracle HCM Cloud. The parameter's value is automatically updated after each successful run.
- c. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

 Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your FTP server and check for the new hire information fetched from your Oracle HCM Cloud instance.

Related Documentation

- [Using the Oracle HCM Cloud Adapter with Oracle Integration 3](#)
- [Using the FTP Adapter with Oracle Integration 3](#)

Export Oracle NetSuite Records to QuickBooks

Use this recipe to export customer and company records from Oracle NetSuite to QuickBooks.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to use Oracle Integration to export customer and company records from Oracle NetSuite to QuickBooks.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. This recipe creates customers and companies in QuickBooks for customers and companies in Oracle NetSuite. Once the QuickBooks customer or company is created, the QuickBooks record ID is mapped with Oracle NetSuite record ID. If the mapping fails, the recipe deletes the QuickBooks record by updating the active state as false.

Note

Ensure that an Oracle NetSuite record must not already have a QuickBooks record ID assigned to it.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- Oracle NetSuite
- An account on Oracle NetSuite with the Administrator role
- QuickBooks
- An account on QuickBooks with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite and QuickBooks instances in order to successfully connect to these external systems using Oracle Integration and export records from Oracle NetSuite to QuickBooks.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration and export records, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See *Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3*.
2. Perform the TBA-related configuration tasks. See *Prerequisites for the Token-Based Authentication Security Policy in Using the Oracle NetSuite Adapter with Oracle Integration 3*.

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
List	Currency	Full
List	Customers	Full
List	Subsidiaries	Full
List	Contacts	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Configure saved search.
 - a. Go to **List**, click **Search** and select **Saved Searches**.

- b. Create a new search or use an existing saved search based on your business use case. See [Defining a Saved Search](#).
- c. In the new saved search, select **Customer** as record type.
- d. Specify the search title and configure a search criteria such that the QuickBooks customer ID field must be empty. See [Advanced Search Criteria Filters](#).
- e. Click **Submit**.

Configure QuickBooks

To access the QuickBooks instance from Oracle Integration and export records, you'll need to create a user account for Oracle Integration on QuickBooks and make a few other configurations. See Prerequisites for Creating a Connection in *Using the QuickBooks Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, <code>https://webservices.netsuite.com/wsdls/v2022_2_0/netsuite.wsdl</code> . See Configure Oracle NetSuite .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier.

Note

You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the QuickBooks Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
QuickBooks Environment	Select Production .

3. In the Security section, enter the following details:

Field	Information to Enter
Client id	Enter the Client ID obtained when you configured QuickBooks. See Configure QuickBooks .
Client secret	Enter the Client Secret obtained when you configured QuickBooks.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

You'll now be redirected to the QuickBooks sign in page.

7. Enter your QuickBooks account credentials.

A page is displayed asking for permission to interact with the account.

8. Click **Yes**.

You're informed that access is allowed. You can now switch back to the QuickBooks Connection window of Oracle Integration to test your connection.

9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

10. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

- b.** On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your QuickBooks instance and check for new customers.
 - a. To view the list of customers, from the QuickBooks homepage, select **Sales** and choose **Customers**.
 - b. To view the list of companies, go to **Settings** on the QuickBooks home page, and under **Profile** select **Switch Company**.

Related Documentation

- *Using the Oracle NetSuite Adapter with Oracle Integration 3*
- *Using the QuickBooks Adapter with Oracle Integration 3*

Export Purchase Orders from Oracle ERP Cloud to FTP Server

Use this recipe to export details of purchase orders created in Oracle ERP Cloud to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe exports a purchase order from Oracle ERP Cloud, maps its values to a comma separated file (CSV) file in Oracle Integration, and uploads the CSV file with the purchase order values to an FTP server.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When a new purchase order is created in Oracle ERP Cloud, the integration flow of the recipe is triggered and it fetches the details of the newly created purchase order from Oracle ERP Cloud. Subsequently, the recipe matches the values in the purchase order to fields in a CSV file, and uploads the CSV file to a specified location in an FTP server. For the first run, the recipe creates the CSV file and adds the purchase order values to it. For subsequent runs, it appends a line with the newly created purchase order values to the same file.

The CSV file contains the following fields:

- Creation Date
- Order Number

- Supplier
- Buyer Name
- Currency
- Total Amount

Only the Order Number field is mandatory. The rest of the fields are optional.

Note that you can upload the CSV file on File Server, an embedded sFTP server within Oracle Integration. To use File Server, you must enable it for your instance. See *Enable File Server in Using File Server in Oracle Integration 3*.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- A secure FTP (sFTP) server or File Server
- An FTP client to access the external sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance and FTP server in order to successfully connect to these external systems using Oracle Integration, and export purchase orders.

Configure Oracle ERP Cloud

To access Oracle ERP Cloud from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this 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 3*.
 - Integration Specialist
 - AttachmentsUser
 - FND_MANAGE_CATALOG_SERVICE_PRIV
3. Create a CSF (Credential Set Framework) key for Oracle ERP Cloud to use when calling the integration. See *Create a CSF Key in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Note

To create the CSF key, the Oracle Integration user must be assigned the ServiceInvoker role. See Oracle Integration Service Roles in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Access Your FTP Server

Obtain an FTP server and ensure that you can access it.

1. Login to the server using your username and password through an FTP client; for example, FileZilla.
2. Create a directory on the server to store the CSV file. Note down the path of the directory.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the URL to access your Oracle ERP Cloud instance in the **ERP Cloud Host** field.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the FTP Connection

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. In the navigation pane, click **Settings**, then **File Server**, then **Settings**.

- b. From the **General** section, copy the IP and port.
2. In the Connections section, click the connection name.
3. In the Properties section, enter the following details:
 - a. **FTP Server Host Address:** the IP for your external FTP file server or the IP you copied from the File Server settings. For instance, server123.company_name.com.
 - b. **FTP Server Port:** the port for your external FTP file server or the port you copied from the File Server settings. For instance, 21.
 - c. **SFTP Connection:** specify the value as No.
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 FTP server.
Password	Enter the password to connect to your FTP server.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the values of the lookup keys in the **ORCL-BRT-FTP_FOLDER** lookup table.

1. In the **Lookup** section, click the lookup name.
2. In the **Property_Value** column for **directory**, replace /InputDirectory with the path to the directory where you want to store the CSV file. See [Access Your FTP Server](#).

If you are using File Server, enter the path that you wrote down when you enabled File Server for your user.
3. In the **Property_Value** for **filename** replace the default value CVS with the name you want to use for the file that stores the purchase order data. For example: PurchaseOrders.csv
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle ERP Cloud instance.
 - b. Create a purchase order.

Creating a purchase order in Oracle ERP Cloud triggers the integration flow of the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to the FTP server/File server and check if the purchase order data have been exported to the CSV file.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*
- *Using File Server in Oracle Integration 3*

Export Workday Employees to an FTP Server

Use this recipe to export employee records from Workday to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe exports Workday employee records to an FTP server, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the employee records from the Workday instance and exports them to an FTP server. While running the recipe, you can configure the integration flow to read Workday employee records created after a specified time, thereby picking up only the new or required employee records in each run. Basic data such as name, ID, date of birth, email, and so on are synchronized between Workday and FTP.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Workday
- An account on Workday with the Administrator role
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your Workday instance and FTP server in order to successfully connect to these external systems using Oracle Integration and export employee records.

Configure Workday

To configure Workday, see Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.

Access Your FTP Server and Create a File

Obtain an SFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server to export the files. Note the path of this directory.
3. Create a file in the directory to write the employee records. Note the name of this file.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Workday Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the Workday host name. For example: <code>https://wd2-impl-services1.workday.com</code>
Tenant Name	Enter the tenant's name of the account created for Oracle Integration on Workday. See Configure Workday .

3. In the Security section, enter the following details:

Field	Information to Enter
Workday Integration User	Enter the username of the account created for Oracle Integration on Workday. See Configure Workday .
Password	Enter the password of the account created for Oracle Integration on Workday.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Public Key Authentication .
Username	Enter the user name to connect to your sFTP server.
Private Key File	Upload the private key file for your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Integration Flow (Optional)

If you want to fetch additional information about Workday employee records (other than what is pre-configured in this recipe), you can modify the filter criteria of the Workday request-mapper element in the integration flow. To successfully process the additional fields fetched, you must update the Stage File element of the integration flow with a new schema file that contains the structure of the updated data set.

Modify the Filter Criteria

1. In the Integrations section, select the **Workday FTP Export Employee Records** integration flow.
2. On the integration canvas, click the **Map to GetWorkers** element, and then click **Actions**  and select **Edit** to open the mapper.
3. In the mapper, under **Response_Group**, find the required element. For example, to fetch employment information, find and click the **Include_Employment_Information** element.
4. In the editor that opens, set:
 - **True**: to export selected information to the FTP server.
 - **False**: to exclude selected information.
5. Click **Save** and then click **Validate**.

To return to the project workspace, click **Go back** .

Update the Stage File Element

1. Add the required additional fields to your sample schema file.

Note

To obtain a sample schema file, run the recipe without modifying the filter criteria initially. The recipe populates the employee data into the specified file on your FTP server. You can use this file as the sample schema file.

2. Edit the **Stage File** element on the integration canvas and upload the modified file. See *Configure a Stage File Action in Using Integrations in Oracle Integration 3*.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the property values for the integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
RecordCount	This integration property holds the number of records to be displayed in one page during integration run time. The default value for RecordCount is 100.
EmailTo	This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, configure the following parameters, and click **Run**.
 - `schedule_para_FileName`: Specify the name of the file to which you want to extract the employee records. See [Access Your FTP Server and Create a File](#).
 - `schedule_para_LastRunTime`: This parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Workday employee records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the employee records from the beginning, set the value as zero.

- `schedule_para_FolderName` : Specify the path of the folder which contains the file with employee details. See [Access Your FTP Server and Create a File](#).

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your FTP server and check for the new employees fetched from your Workday instance.

Related Documentation

- *Using the Workday Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Extract Approved Expense Reports from SAP Concur and Write to an FTP Server

Use this recipe to extract approved expense reports from SAP Concur and write to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe extracts approved expense reports from SAP Concur and write them to an FTP server, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the approved expense reports from SAP Concur and writes them to the specified location on the FTP server.

System and Access Requirements

- Oracle Integration 3
- A valid account on SAP Concur
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your SAP Concur and FTP instances to successfully connect to these external systems using Oracle Integration.

Configure SAP Concur

To configure SAP Concur, see Prerequisites for Creating a Connection in *Using the SAP Concur Adapter with Oracle Integration 3*.

Access Your FTP Server and Create a Directory

Obtain an SFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server where the approved expense reports from SAP Concur will be written.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the SAP Concur Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Concur Instance Url	Enter the URL for the SAP Concur application host location.
Edition	Select Professional Edition .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Resource Owner Password Credentials selected.
Client Id	Enter the client id generated for Oracle Integration on the SAP Concur instance. See Configure SAP Concur .
Client Secret	Enter the client secret generated for Oracle Integration on the SAP Concur instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Name	Enter the host address of your sFTP server.
FTP Server Port	Enter the server port number.
SFTP Connection	Select True .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the username to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration property values. You can update these values only after activating the integration flow.
 - a. **EmailTo**: This integration property holds the email address to which runtime exception emails are sent. Enter an email address of your choice.
 - b. **RetryCount**: This integration property holds the number of retries to attempt if a record fails while importing employee records at run time. Enter a number of your choice.
 - c. **Filename**: The name of the file that stores the expense reports.
 - d. **waitTime**: This integration property holds the wait time before retrying a failed operation.
 - e. **Directory**: The directory path where the expense reports will be uploaded in the FTP server.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your SFTP server and check if the extracted approved expense records from SAP Concur are written in the specified directory.

Related Documentation

- *Using the SAP Concur Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Extract Data from Unstructured Files Using OCI Vision

Use this recipe to read and extract data from unstructured files such as PDF, JPG, PNG or TIFF using Oracle Cloud Infrastructure Vision (OCI Vision).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe extracts data from unstructured files such as image files and PDFs stored in an FTP server using OCI Vision, and then rewrites the processed files to the FTP server.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered the integration flow reads the files from the FTP server, extracts relevant data using OCI Vision, and then renames and writes the processed files back to the FTP server. Note that the recipe can run a maximum of 30 files at a time.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Oracle Cloud Infrastructure (OCI)
- OCI Vision
- A federated user account in OCI
- A secure FTP (SFTP) server
- An FTP client to access the FTP server

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and OCI instance in order to successfully connect to these external systems using Oracle Integration and extract data from unstructured files.

1. Access your FTP Server and upload files that are to be processed.
 - a. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
 - b. Create a directory on the server from which to read the files. Note down the path of this directory.
 - c. Upload sample files into the directory you've created. Note down the format of the files.
2. Log in to your OCI instance as an **Administrator** and perform the following tasks.

For more information, refer to the Oracle Cloud Infrastructure documentation.

 - a. Create a compartment.
 - b. Create a federated user.
 - c. Assign a role to the federated user.
 - d. Create a local user.
 - e. Create a user group.
 - f. Create a policy.
 - g. Obtain the following for creating a connection to OCI Vision from Oracle Integration. See section **OCI Signature Version 1 Security Policy Use** in Prerequisites for Creating a Connection in *Using the REST Adapter with Oracle Integration 3*.
 - i. Get your Tenancy and User OCIDs.
 - ii. Create your own private key or generate it from the Oracle Cloud Console.
 - iii. Get the key's fingerprint.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Vision API Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .

Field	Information to Enter
Connection URL	Enter the endpoint URL of your OCI Vision instance. For example: <code>https://vision.aiservice.<tenancy location>.oci.oraclecloud.com/20220125</code>
3. In the Security section, enter the following details:
- | Field | Information to Enter |
|--------------|--|
| Tenancy OCID | Enter the tenancy OCID that you obtained.
See Before You Install the Recipe . |
| User OCID | Enter the user OCID that you obtained. |
| Private Key | Click Upload and select the API key generated earlier. |
- Note**

Before you upload the private key, you must convert it into the PKCS1 format.
Field	Information to Enter
Finger Print	Enter the finger print that was generated for the key.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
- A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the FTP Vision Invoice Rename Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:
- | Field | Information to Enter |
|-------------------------|---|
| FTP Server Host Address | Enter the host address of your sFTP server. |
| FTP Server Port | Enter 22. |
| SFTP Connection | Select Yes from the list. |
3. In the Security section, enter the following details:
- | Field | Information to Enter |
|-----------------|---|
| Security Policy | Select FTP Server Access Policy . |
| User Name | Enter the user name to connect to your sFTP server. |
| Password | Enter the password to connect to your sFTP server. |
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
- A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **filePattern**: This integration property holds the format of the files to be read from the FTP server. For example: *.pdf
 - b. **directory**: This integration property holds the path of the directory on the FTP server from where the files must be read. For example: /user2/InvoiceDemo.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your FTP server, and check if the processed files have been updated.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- [Oracle Infrastructure Cloud Documentation](#)

Extract Document Information using OCI AI Services

Use this recipe to extract information from uploaded documents, such as a passport or driving license using the artificial intelligence capabilities of the following OCI AI service - Oracle Cloud Infrastructure Document Understanding (OCI Document Understanding).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe uses the OCI Document Understanding service to automate the extraction of key data fields from uploaded identity documents, specifically passports and driving licenses.

When you upload a document, the flow first classifies the document type using OCI Document Understanding REST APIs. Based on the identified document type (Passport or Driving License), the Oracle Integration native action, **OCI Document Understanding** is used to parse and extract structured data such as name, date of birth, document number, issue date, expiry date, and so on. The extracted data can then be used for downstream processing, such as identity verification, user onboarding, or storage in a database.

System and Access Requirements

- Oracle Integration 3 version 25.04 or higher
- OCI Document Understanding

Before You Install the Recipe

You must perform the following configuration tasks in Oracle Cloud Infrastructure. To call the OCI Document Understanding REST APIs, OCI Signature Version 1 Security Policy is used in the REST Adapter.

- To create the signing key, obtain the tenancy OCID and user OCID details, see [OCI Signature Version 1 Security Policy Use](#) in *Using the REST Adapter with Oracle Integration 3*.

Install and Configure the Recipe

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the OCI Document Understanding Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the **Connection URL** as `https://document.aiservice.<oci_region_identifier>.oci.oraclecloud.com`. See [Regions and Availability Domains](#) in *Oracle Cloud Infrastructure Cloud Adoption Framework* to find the OCI region identifier.
3. In the Security section, enter the following details: noted when you completed the steps in [Before You Install the Recipe](#).

Field	Information to Enter
Security Policy	Select OCI Signature Version 1 .
Tenancy OCID	Enter the OCI tenancy OCID.
User OCID	Enter the OCI user OCID.
Private Key	Upload the private key.
Finger Print	Enter the fingerprint.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Extract Document Information using AI** integration flow, then select **Run**. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click the **Body** tab, choose **File** option and upload the document such as Passport and Driving License.
 - c. Click **Run**

Success Response of the flow:

```
{
  "status" : "Success",
  "documentType" : "DRIVER_LICENSE",
  "firstName" : "JUDY JANE",
  "lastName" : "DOE",
  "gender" : "F",
  "dateOfBirth" : "01/01/1997",
  "documentId" : "G1111111",
  "documentExpiryDate" : ""
}
```

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify that the information is extracted correctly as per the document uploaded.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- [OCI Document Understanding Service REST APIs](#)

Fetch Data in Chunks from Oracle Autonomous Transaction Processing

Use this recipe to fetch data in small chunks from Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to use *stored procedures* to fetch supplier data in small chunks from Oracle Autonomous Transaction Processing. You can use the recipe to implement use cases where due to service limits data cannot be fetched in bulk, but have to be fetched in smaller chunks.

To use the recipe, you must install the recipe and configure the connections and other resources within it.

The recipe uses a database package **XX_SUPPLIERS_PAGINATION_PKG**. The package uses a PL/SQL procedure **GET_SUPPLIERS_PAGINATION_PROC** which accepts the following input parameters:

- **i_limit**: A numeric parameter that specifies how many records are to be fetched from the ATP database at a time.
- **i_page**: A numeric parameter that specifies which page's data have to be fetched.

System and Access Requirements

- Oracle Integration 3
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Before You Install the Recipe

Log in to your ATP database instance as an **Administrator** and perform the following tasks.

1. Perform general configuration tasks. See Prerequisites for Creating a Connection in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.
2. Create necessary objects.

To create necessary objects, run the following script:

```

CREATE TABLE "ADMIN"."SUPPLIERS"
(
  "ID" NUMBER,
  "SUPPLIER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "SUPPLIER_NUMBER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "TAX_ORGANIZATION_TYPE_CODE" VARCHAR2(200 BYTE) COLLATE
"USING_NLS_COMP",
  "TAX_ORGANIZATION_TYPE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "SUPPLIER_TYPE_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "SUPPLIER_STATUS" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "BUSINESS_RELATIONSHIP_CODE" VARCHAR2(200 BYTE) COLLATE
"USING_NLS_COMP",
  "BUSINESS_RELATIONSHIP" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "TAX_REGISTRATION_COUNTRY" VARCHAR2(200 BYTE) COLLATE
"USING_NLS_COMP",
  "TAXPAYER_COUNTRY_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "TAXPAYER_COUNTRY" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "TAXPAYER_ID" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "CREATION_DATE" TIMESTAMP (6) DEFAULT ON NULL SYSDATE,
  "UPDATION_DATE" TIMESTAMP (6) DEFAULT ON NULL SYSDATE
) DEFAULT COLLATION "USING_NLS_COMP" ;

CREATE UNIQUE INDEX "ADMIN"."SUPPLIERS_PK" ON "ADMIN"."SUPPLIERS" ("ID")
;

ALTER TABLE "ADMIN"."SUPPLIERS" ADD CONSTRAINT "SUPPLIERS_PK" PRIMARY KEY
("ID")
  USING INDEX ENABLE;

ALTER TABLE "ADMIN"."SUPPLIERS" MODIFY ("CREATION_DATE" NOT NULL ENABLE);

ALTER TABLE "ADMIN"."SUPPLIERS" MODIFY ("UPDATION_DATE" NOT NULL ENABLE);

CREATE OR REPLACE EDITIONABLE TRIGGER "ADMIN"."SUPPLIERS_TRIGGER"
BEFORE INSERT ON SUPPLIERS
FOR EACH ROW
  WHEN (new.ID IS NULL) BEGIN
    :new.ID := SUPPLIERS_SEQ.NEXTVAL;
  END;
/
ALTER TRIGGER "ADMIN"."SUPPLIERS_TRIGGER" ENABLE;

CREATE OR REPLACE EDITIONABLE TRIGGER "ADMIN"."SUPPLIERS_UPDATE_DATE"
BEFORE UPDATE ON SUPPLIERS
FOR EACH ROW
BEGIN
  :NEW.UPDATION_DATE := SYSDATE;
END;

CREATE OR REPLACE EDITIONABLE PACKAGE
"ADMIN"."XX_SUPPLIERS_PAGINATION_PKG" AS
  PROCEDURE GET_SUPPLIERS_PAGINATION_PROC (
    I_PAGE      IN NUMBER,
    I_LIMIT     IN NUMBER,
    I_PAGE_SIZE OUT NUMBER,

```



```

        P_DATA      OUT SYS_REFCURSOR
    );

END XX_SUPPLIERS_PAGINATION_PKG;
/

CREATE OR REPLACE EDITIONABLE PACKAGE BODY
"ADMIN"."XX_SUPPLIERS_PAGINATION_PKG" AS

    PROCEDURE GET_SUPPLIERS_PAGINATION_PROC (
        I_PAGE      IN NUMBER,
        I_LIMIT     IN NUMBER,
        I_PAGE_SIZE  OUT NUMBER,
        P_DATA      OUT SYS_REFCURSOR
    ) AS
        COUNT1 NUMBER;
    BEGIN
        SELECT
            COUNT(*)
        INTO COUNT1
        FROM
            SUPPLIERS;

        I_PAGE_SIZE := CEIL(COUNT1 / I_LIMIT);
        OPEN P_DATA FOR SELECT
                                *
                                FROM
                                    SUPPLIERS
                                ORDER BY
                                    ID
                                OFFSET NVL((I_PAGE - 1), 0) * I_LIMIT ROWS FETCH
    NEXT I_LIMIT ROWS ONLY;

    END;

END XX_SUPPLIERS_PAGINATION_PKG;

```

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See [Before You Install the Recipe](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Select the check box, then click Upload to upload the wallet file. See Before You Install the Recipe .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Extract Data from ATP in Chunk** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify that data is fetched from Oracle Autonomous Transaction Processing in chunks according to the parameters defined in the stored procedure.

Related Documentation

- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Fusion HCM License and Certification Validation by EverCheck

Use this recipe to import employees that are verified in EverCheck to Oracle HCM Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe verifies employee licenses, certificates, and registrations in EverCheck and imports the verified employee data into Oracle HCM Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe provides the following key capabilities:

- Extracts employee licenses, certificates, and registrations data in bulk from Oracle HCM Cloud using HCM Extracts.
- Transforms the extracted data into EverCheck file format.
- Transfers the transformed file to an EverCheck sFTP server.
- Retrieves the file containing processed and monitored employee licenses, certificates and registrations data from Evercheck sFTP server.
- Transforms the retrieved file containing processed employee data into HCM Data Loader dat file format.
- Imports the employee licenses, certificates, and registration verification status into Oracle HCM Cloud using HCM Data Loader.

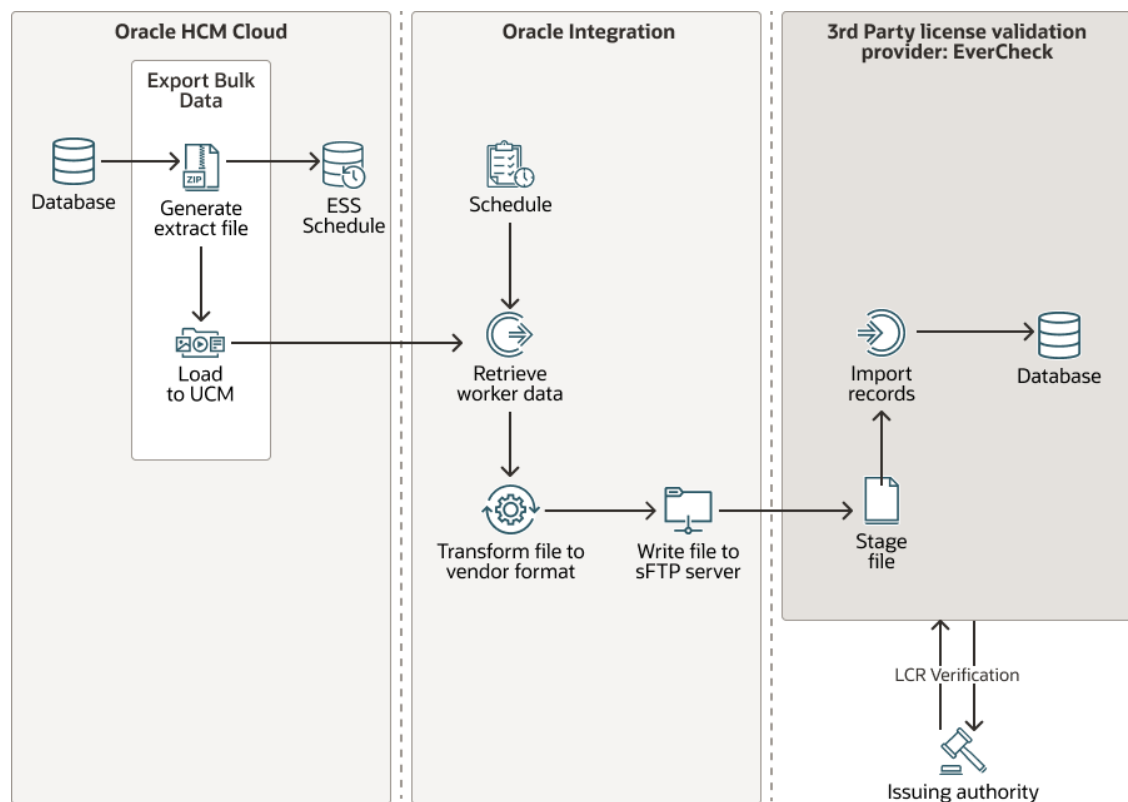
System and Access Requirements

- Oracle Integration, Version 24.06 or higher
- An account in Oracle Integration with the Service Invoker role
- An account in Oracle Integration with the Developer role
- Oracle HCM Cloud, Version 24B or higher
- EverCheck License, Certifications, and Registrations Verifications Monitoring service agreement

Recipe Schema

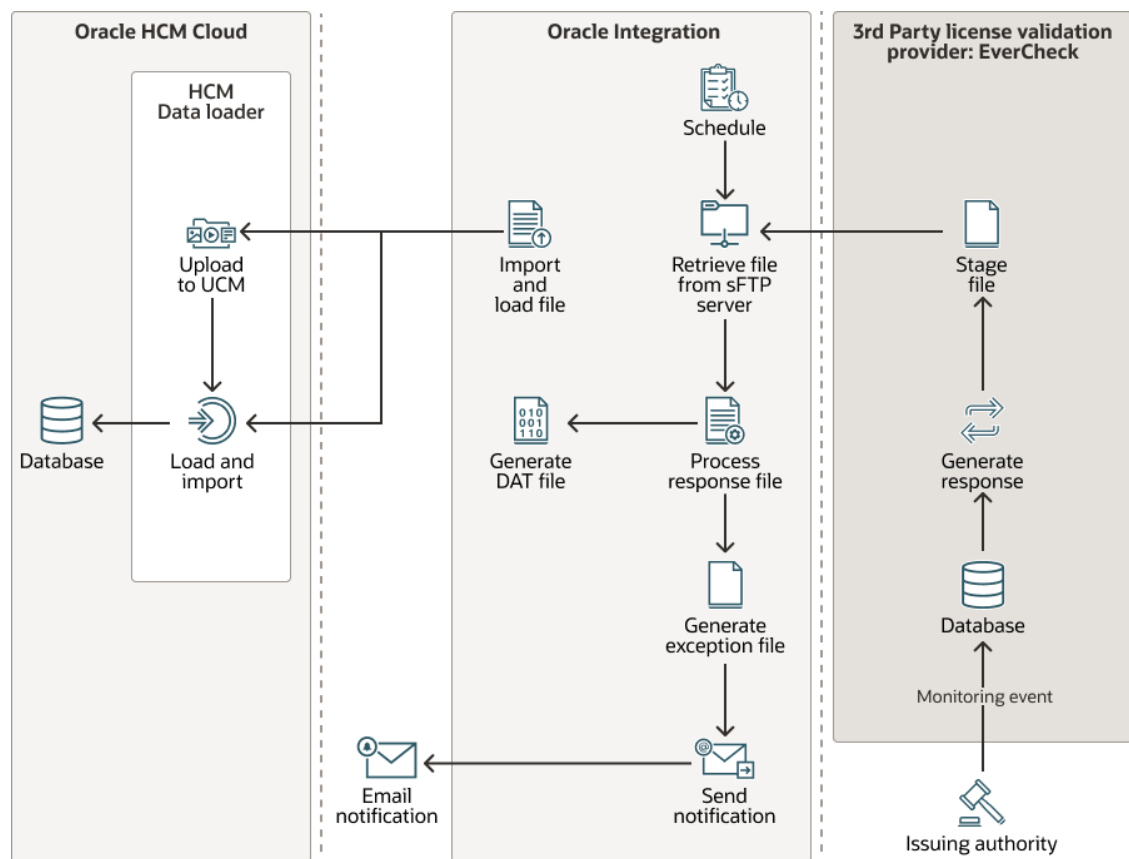
This section provides an architectural overview of the integration flows in the recipe

Oracle HCM Outbound Integration to EverCheck



In Oracle HCM Cloud, HCM Extract generates a file on a predefined schedule containing all of the employees required to be monitored. This file is retrieved by a scheduled integration running on Oracle Integration. The file is transformed from XML to CSV as part of the execution of the integration flow, and the resulting file is placed on to an sFTP EverCheck server. This file will be used to generate the current active monitoring roster for EverCheck's license monitoring service.

Oracle HCM Inbound Integration from EverCheck



On a scheduled basis, EverCheck generates a file containing updates to licenses, certifications, and registrations which it is monitoring. These updates are generated from the various issuing authorities which are ultimately the primary source for licenses, certifications, and registrations. This file is retrieved by a scheduled integration running on Oracle Integration. The file is transformed into a DAT file format. As an optional step, a filter can be applied in which certain statuses are separated for manual review. You can also configure a lookup table to translate issuing authority statuses into seeded HCM Certification status. Both the filter and the lookup table are configurable. Subsequently, the DAT is loaded into Oracle HCM Cloud via the HCM Data Loader.

Assumptions

- Based on your requirements, you may enhance the recipe to validate or selectively filter employees which should not be created in EverCheck.
- The integration flows of the recipe include the minimal data mapping required to synchronize the data with EverCheck. You can have additional mappings based on your specific requirements. The final section of this document details the mapping between the various file types of these integrations. For details of the mapping between the various file types of the integrations, see [Bulk File Data Models](#).
- The integration flows of the recipe include sample error handling and notifications. A robust error handling based on your own environment can be additionally implemented.
- This recipe does not include a SystemUser integration between Oracle HCM Cloud and EverCheck. The setup of EverCheck system and manager users will need to be done directly in EverCheck via the existing UI or a separate integration will have to be created.

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud instance and EverCheck instance in order to connect to these external systems using Oracle Integration and successfully import employees verified in EverCheck into Oracle HCM Cloud.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration, you'll require a separate user account on Oracle HCM Cloud if you do not already have one in place.

Create a separate integration user account in Oracle HCM Cloud to access Oracle HCM Cloud from Oracle Integration.

Log in to the Oracle HCM Cloud instance as an **Administrator** and perform the following tasks:

1. Create a user account for Oracle Integration which will be required to connect to Oracle HCM Cloud from Oracle Integration using the Oracle HCM Cloud adapter.
2. Assign roles to the user account. See *Assign Required Roles to an Integration User in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

For more information, see *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure EverCheck

To access EverCheck from Oracle Integration, you must perform certain configurations on your EverCheck instance. In addition, you'll require a separate user account and sFTP site with EverCheck with necessary security grants. Due to the nature of the services offered by EverCheck, a service implementation plan has to be undertaken between yourself and EverCheck. This will ensure successful scope and implementation of EverCheck's services.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

- [Configure the Oracle HCM Cloud Adapter Connection](#)
- [Configure the FTP Connection](#)
- [Configure the Fusion HCM Cloud REST Connection](#)
- [Configure the Lookup Table](#)
- [Configure the Oracle HCM Outbound Integration to EverCheck](#)
- [Configure the Oracle HCM Inbound Integration from EverCheck](#)
- [Bulk File Data Models](#)

Configure the Oracle HCM Cloud Adapter Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on the Oracle HCM Cloud instance. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle HCM Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of EverCheck's sFTP server.
FTP Server Port	Enter the port number of EverCheck's sFTP server.
SFTP Connection	Select Yes .

3. In the Security section, enter the following details:

Field	Information to Enter
Security policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to EverCheck's sFTP server.
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

For information on your EverCheck sFTP configuration, contact your EverCheck implementation specialist.

Configure the Fusion HCM Cloud REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the Oracle HCM Cloud host name. For example: <code>https:// your_domain_name.fa.DC.oraclecloud.com</code>

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of the account created for Oracle Integration on the Oracle HCM Cloud instance. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle HCM Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

This recipe includes a lookup table called **everCheckStatusLookUp** which is used to translate license statuses that EverCheck captures from the issuing authorities and sends to the Oracle HCM Inbound integration, to setup statuses in your Oracle HCM Cloud instance. Based on the nature of your business, the values can be different, so you can configure this lookup table according to your needs.

1. In the Lookups section, click the lookup name.
2. If required, update and map the values in the **everCheckStatus** column to corresponding values in the **hcmLicenseStatus** column.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Oracle HCM Outbound Integration to EverCheck

You must configure the **Oracle HCM Outbound Integration to EverCheck** integration flow to successfully import employee details of Oracle HCM Cloud in bulk that are verified in EverCheck.

Before configuring the Oracle HCM Outbound Integration to EverCheck integration flow, ensure that you have extracted employee data from Oracle HCM Cloud in bulk using the HCM Extract tool. For more information about the tool, refer to the [HCM Extracts](#) guide.

You can extract the following employee information:

- Basic details of the employee such as Person Number, First Name, Last Name, Person Email Address, Phone Number, and Home Address.
- Any licenses, certifications and registrations associated to the *job* assigned to the employee.
- Any licenses, certifications and registrations associated to the *position* assigned to the employee.

The following high-level steps are required to configure the Oracle HCM Outbound Integration to EverCheck integration flow:

1. [Configure the Oracle HCM Cloud Instance Extract](#)
2. [Submit an Extract](#)
3. [Schedule Extract Submission](#)
4. [Configure the Integration Variables](#)
5. [Change the Extract Configuration](#)
6. [Setup Flex Fields within EverCheck](#)

Configure the Oracle HCM Cloud Instance Extract

Upload the pre-built extract to the Oracle HCM Cloud instance, update it (if required) and configure it. This is a recommended step.

1. Log in to the Oracle HCM Cloud instance.
2. Navigate to **Manage HCM Extract Definitions**.

From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Extract Definitions** or **Setup and Maintenance**, and then select **Manage HCM Extract Definitions**.

3. On the Extract Definitions page, click **Import**.
4. Click **OK**, then click **OK** again.
5. Next click **Edit** to the newly created extracted.
6. Select **Extract Execution Tree** and click **Compile All Formula**.
7. Refresh the screen and make sure all the fast formula status show passed status.
8. Click **Export XML Schema** to download the .xsd file.
9. Click **Save and Close**.

Submit an Extract

You have to submit the HCM Extract in order to configure the Oracle HCM Outbound Integration to EverCheck integration flow in Oracle Integration. The integration will use this submitted extraction for fetching data while running.

1. From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Submit Extract**.
2. Search for the extract with its name.
3. Click the extract.
4. Enter the extract instance name and select the effective date.

Schedule Extract Submission

You must schedule an extract submission to fetch the new data per the required frequency for running the Oracle HCM Outbound Integration to EverCheck integration flow in Oracle Integration.

1. From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Submit Extract**.
2. Search for the extract with its name.
3. Click the extract.
4. Enter the extract instance name and effective date.
5. Select the required frequency for extract submission, and select the start and end date.

Note

Although you can set any value as minimum frequency, the best practice is to use the extractsubmission frequency as greater than or equal to daily. The recommended value is daily. This should be agreed with your EverCheck implementation specialist.

6. Click **Submit**.

Configure the Integration Variables

Update the integration variables based on your environment and business rules.

1. In the Integration section of the project workspace in Oracle Integration, click the **Oracle HCM Outbound Integration to EverCheck** integration.
2. Click the **AssignGlobalParams** Assign action, and configure the following:

Field	Information to Enter
Constant_OutputFileName	By default, EverCheck requires the file to have the following naming convention for Oracle HCM Outbound records: nnn_CustomEmployee.csv. Here, nnn will be replaced by your organization's 3 digit id. Validate this with your EverCheck implementation specialist.

Field	Information to Enter
ftpDropOffFolder	This is a folder where nnn_CustomerEmployee.csv file will be written to on the EverCheck sFTP server. By default EverCheck uses the /CustomFile folder for inbound files. Validate this with your EverCheck implementation specialist.

Note that other values set in the **AssignGlobalParams** Assign action should be left as is, since they are referenced throughout the recipe.

Change the Extract Configuration

In order for the appropriate HCM Extract to be linked to this recipe, it is necessary to update the extract configuration.

1. In the Integration section of the project workspace in Oracle Integration, click the **Oracle HCM Outbound Integration to EverCheck** integration.
2. Click the Invoke element **HCMExtract**, and select **Edit** from the menu.
3. Click **Operations** and update the integration name. The integration name should be same as the name added in the HCM Extract while configuring the Oracle HCM Cloud instance extract in Oracle HCM Cloud.
4. Click **Done**.
5. Save the changes made in the integration.

Setup Flex Fields within EverCheck

You have to set up the following fields in the EverCheck instance. These fields will be used by the recipe for the successful implementation of your integration.

- CertificationId
- ProfileId
- Employee Residence State

Contact your EverCheck implementation specialist for more details and also for assistance in setting up these fields.

Configure the Oracle HCM Inbound Integration from EverCheck

The **Oracle HCM Inbound Integration from EverCheck** integration flow retrieves the file with the verified employee licenses, registrations and certifications data from EverCheck. This file is transformed to DAT file format in Oracle Integration. As an optional step, a filter can be applied in which certain statuses are separated for manual review. You can also configure a lookup table to translate Issuing Authority statuses into seeded HCM Certification status. Subsequently, the DAT file with the details is uploaded into Oracle HCM Cloud using the HCM Data Loader.

The following high-level steps are required to configure the Oracle HCM Inbound Integration from EverCheck integration flow:

1. [Configure the Integration Variables](#)
2. [Configure the Filter Conditions in the Integration](#)
3. [Configure the Use of the Lookup Table in the Integration](#)

Configure the Integration Variables

1. In the Integration section of the project workspace in Oracle Integration, click the **Oracle HCM Inbound Integration from EverCheck** integration.
2. Click the **AssignGlobalParams** Assign action, and configure the following:

Field	Information to Enter
ftpPickUpFolder	By default EverCheck uses the compliancecheck folder for processed records. Validate this with your EverCheck implementation specialist. Validate this with your EverCheck implementation specialist.
ftpDropOffFolder	Exception reports will be written in this folder. These records are returned from EverCheck, but are not automatically processed and loaded into Oracle HCM Cloud. They have to be validated. Specify the folder location on your SFTP where the file can be placed for further processing or manual review.
zipFileName	This value refers to the name of the zip file containing the dat file being loaded into UCM and processed by HCM Data Loader. It can be left as is or changed based on your preference. The recipe is configured to return the file name dateTime format. For example: talentCertification_2024-02-12T16_00_34.434Z.zip.
zipIssuesFileName	This value refers to the name of the exception records file being written out as part of the recipe. It can be left as is or changed based on your preference. The recipe is configured to return the file name dateTime format. For example: employeeLicenseIssues_2024-02-12T17_39_04.931Z.zip

Note that other values set in the **AssignGlobalParams** action should be left as is, since they are referenced throughout the recipe.

Configure the Filter Conditions in the Integration

The default filter condition configured within the recipe allows for separate processing of licenses, certifications, and registrations into two different work streams.

If values returned from EverCheck indicate a license is Active or Current, then the license record is processed and loaded into Oracle HCM Cloud. Any other values will result in the license record being written to the Exception Notification file and sent to an SFTP server.

You can edit the filter conditions to implement your business requirements.

1. In the Integration section of the project workspace in Oracle Integration, click the **Oracle HCM Inbound Integration from EverCheck** integration.
2. Click the For-Each action, and expand the Switch action under it.
3. Click **Route 1**, and then click **Actions . . .** and select **Edit**.
4. In the Configure route panel, you can click an expression and edit it if required.

Configure the Use of the Lookup Table in the Integration

Within the integration mapping as part of the generation of a DAT file, the existing mappings received from EverCheck are mapped to Oracle HCM Cloud values. This is done using the lookup table configured in the recipe. See [Configure the Lookup Table](#).

The mapping occurs in the **Map to writeTempEmployeeLicenseFile** element.

1. In the Integration section of the project workspace in Oracle Integration, click the **Oracle HCM Inbound Integration from EverCheck** integration.
2. Click the For-Each action, and expand the Switch action under it.
3. Click the **Map to writeTempEmployeeLicenseFile** element, and then click **Actions** and select **Edit** to open the mapper.
4. On the target side of the mapping screen, expand the **Ever Check License Employee Record Set Name** until the **License Status** field is shown.
5. Select the **License Status** field to see the expression for the field.
At this point, you can do one of the following, depending on your requirements:
 - Leave as is and leverage the default values of the lookup table within the integration to transform the inbound License Status to reflect the values within Oracle HCM Cloud. This may require additional values to be added to the lookup table and configurations in Oracle HCM Cloud.
 - Remove the lookup table reference and simply map the values which are being returned from EverCheck to the Statuses maintained in Oracle HCM Cloud. To achieve this, clear the Expression box and drag the License Status on the source side to the License Status on the target side. Note that additional configurations may also need to be made in Oracle HCM Cloud.

Bulk File Data Models

This section lists different data mappings between Oracle HCM Cloud and EverCheck. These can be extended based on your specific business requirements and implementation considerations.

Find more information about the data mappings in the following topics:

- [HCM Extract to EverCheck – Oracle HCM Outbound Integration to EverCheck](#)
- [Oracle HCM Inbound Integration from EverCheck](#)

HCM Extract to EverCheck – Oracle HCM Outbound Integration to EverCheck

Mapping Table

HCM Extract			EverCheck			
#	Field Name	Transformation	Mandatory	Path	Field / Col Name	Instructions/Rules
1	<License_or_Certification_Number>	N	Y	Col1	License Number	License number should be submitted exactly as it appears on the primary source. This includes any instances where a license number contains alpha characters. Leading zeros in the numeric portion of the license number should be included. Alpha, Numeric, Special Characters as needed.
2	<State_Code>	N	Y	Col2	State Indicator	Alpha
3	<Person_First_Name>	N	Y	Col3	Employee Last Name	Alpha
4	<Person_Last_Name>	N	Y	Col4	Employee First Name	Alpha
5	<Person_Middle_Name>	N	N	Col5	Employee Middle Name	Optional. Alpha.
6	<Employee_ID>	N	Y	Col6	Employee ID	Should your Employee ID have leading zeros, the zeros will be accepted and displayed. Numeric or Alpha.
7	<Item_ID>	N	Y	Col7	Item ID	Used to describe the profession of the reported license. You may use the existing codes in your HR system to populate this field. Alpha.
8	<License_or_Certification_Name>	N	N	Col8	License Description	Optional. Used to describe the profession of the reported license. You may use the existing descriptions in your HR system to populate this field. Alpha.
9	<Expiration_Date>	format-dateTime(Extract_Assignment_Person_Certifications_Expiration_Date, '[Y0001]-[M01]-[D01]')	N	Col9	Expiration Date	Optional. Used for date-tracked credentials. Alpha, Numeric, Special Characters as needed.
10	<Person_Email_Address>	N	Y	Col10	Employee Email Address	Numeric or Alpha.
11	<National_Identifier_Number>	N	N	Col11	SSN	This is optional but required for some Manual Verification's (ex: ASCP)
12	<Person_Date_of_Birth>	format-dateTime(Person_Date_of_Birth, '[Y0001]-[M01]-[D01]')	N	Col12	DOB	This is optional but required for some Manual Verification's (ex: ASCP)
13	<Certification_ID>	N	Y	Col13	CertificationId	Flex field held on the license
14	<Profile_Id>	N	Y	Col14	ProfileId	Flex field held on the license
15	<Person_Address_Region2>	N	Y	Col15	Employee Residence State	Flex field held on the license

HCM Extract Sample [Full Extract of Licenses and Certifications]

```

<Person_License_From_Job_h>
  <Person_License_From_Job>
    <Person_License_Details>
      <Person_First_Name>Anderson</Person_First_Name>
      <Person_Middle_Name>/>
      <Person_Last_Name>Zoey</Person_Last_Name>
      <National_Identifier_Number>123-45-1253</National_Identifier_Number>
      <Person_Email_Address>abcd.ef20@adcdat05.us.oracle.com</Person_Email_Address>
      <Person_Date_of_Birth>1980-01-01T00:00:00.000Z</Person_Date_of_Birth>
      <Employee_ID>300100590843192</Employee_ID>
      <Profile_ID>300100590844958</Profile_ID>
      <Certification_ID>300100591144060</Certification_ID>
      <License_or_Certification_Number>2002022</License_or_Certification_Number>
      <License_or_Certification_Name>/>
      <Country_Code>US</Country_Code>
      <Issue_Date>/>
      <Issued_By>National Board of Medical Examiner</Issued_By>
      <Expiration_Date>/>
      <State_Code>/>
      <License_Status>/>
      <License_Title>/>
      <Item_ID>CNA</Item_ID>
      <Person_Country>US</Person_Country>
      <Person_Country_Name>United States</Person_Country_Name>
      <Person_Postal_Code>32013</Person_Postal_Code>
      <Person_Address_Line1>1019</Person_Address_Line1>
      <Person_Address_Line2>Line 19</Person_Address_Line2>
      <Person_Address_Line3>Street 19</Person_Address_Line3>
      <Person_Address_Line4>Locality 19</Person_Address_Line4>
      <Person_Address_Region1>Lafayette</Person_Address_Region1>
      <Person_Address_Region2>FL</Person_Address_Region2>
      <Person_Address_Region3>/>
      <Person_Home_Town_or_City>Day</Person_Home_Town_or_City>
    </Person_License_Details>
  </Person_License_From_Job_h>

```


Note

The data shown above is test data and is not actually related to an actual person or licenses.

EverCheck Sample – nnn_CustomEmployee.csv

```
1 License Number,State Indicator,Employee Last Name,Employee First Name,Employee Middle Name,Employee Id,Item Id,License Description,Expiration Date,Employee
  Email Address,Work Group Dept Codes,Work Group Mgr Id Codes,SSN,DOB,CertificationId,ProfileId,Employee Residence State
2 223293,FL,Her,Alberto,,300100587331903,REGNRS,,2024-09-29,abcd.ef23@adcdat05.us.oracle.com,,,1256,1980-01-01,300100587504859,300100587241664,FL
```

Note

The data shown above is test data and is not actually related to an actual person or licenses.

Oracle HCM Inbound Integration from EverCheck**Mapping Table**

#	EverCheck	Field / Col Name	Path	Transformation	HDL DAT File Format	Example Value	Mandatory	Column Name	Column Name	Instructions/Rules
1	Employee ID	Col1	Not mapped							Not required
2	Last Name	Col2	Not mapped							Not required
3	First Name	Col3	Not mapped							Not required
4	State	Col4	Not mapped							Not required
5			Hardcoded		MERGE			METADATA		structural
6			Hardcoded		TalentCertification			TalentCertification		structural
7	License or Certification Number	Col5	Not mapped		OT.12345					Not required
8	CertificationId	Col6	N		300100010474623	Y		CertificationId	CertificationId	
9	ProfileId	Col7	N		300100010487654	Y		ProfileId	ProfileId	
10			Hardcoded		Y	Y		ExternalPartyVerified	ExternalPartyVerified	
11			Hardcoded		EverCheck	Y		LastVerifiedBy	LastVerifiedBy	
12			Sformat() Current date-Time		2024/12/31	Y		LastVerifiedOn	LastVerifiedOn	
13	Item ID	Col8	N		Oracle India Pvt Ltd	Y		IssuedBy	IssuedBy	
14	Issue Date	Col9	Sformat() date-Time		2020/12/20	N		IssueDate	IssueDate	
15	Expiration Date	Col10	Sformat() date-Time		2020/12/20	N		ExpirationDate	ExpirationDate	
16			Sformat() Current date-Time		2020/12/20	N		RenewalDate	RenewalDate	
17			Sformat() Current date-Time		2020/12/20	Y		LastRenewalDate	LastRenewalDate	
18	License Status	Col11	Lookup Table		Active	Y		Status	Status	
19			Hardcoded		Y	Y		VerifiedFlag	VerifiedFlag	

EverCheck Sample (CSV) EC_ExportData.csv

```
1 Employee ID,Last Name,First Name,State,License or Certification Number,CertificationId,ProfileId,Item ID,Issue Date,Expiration Date,License Status
2 300100587478936,KANG,ZOE,FL,411248,300100592940259,300100587483255,DO,10/26/2020,01/18/2024,ACTIVE
```

Note

The data shown above is test data and is not actually related to an actual person or licenses.

HCM Data Loader (DAT) EnhancedTalentProfile.dat

```

1 METADATA|TalentCertification|CertificationId|ProfileId|ExternalPartyVerified|LastVerifiedBy|LastVerifiedOn|IssuedBy|IssueDate|ExpirationDate|Status|VerifiedFlag
2 MERGE|TalentCertification|300100587584859|300100587241664|Y|EverCheck|2024/02/12|EverCheck|2017/01/08|2024/09/29|Valid|Y

```

Note

The data shown above is test data and is not actually related to an actual person or licenses.

Exception Record Report (CSV)

```

1 300100587927470,THOMPSON,BILL,,290101040740876,300100587951623,300100587933387,CPHT,,07/31/2006,UNCERTIFIED
2 300100587478936,KANG,ZOE,FL,101807,300100587584883,300100587483255,ARNP,06/13/1999,05/15/2025,INACTIVE

```

The file structure for the Exception Record Report follows the format of the original inbound EC_ExportData.csv.

Note

The data shown above is test data and is not actually related to an actual person or licenses.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** . . . for each of the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule the integrations to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In your Oracle HCM Cloud instance, check if the verified employee licenses, certificates and registrations have been imported.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Fusion HCM License and Certification Validation by HireRight

Use this recipe to import employees that are verified in HireRight to Oracle HCM Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe verifies employee licenses and certificates in HireRight and imports the verified employee data into Oracle HCM Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe provides the following key capabilities:

- Extracts employee licenses and certificates in bulk from Oracle HCM Cloud using HCM Extracts.
- Transforms extracted details into HireRight REST calls to create the initial roster.
- Extracts delta details daily and transforms the extracted details into HireRight REST calls to manage the roster.
- Implements daily recertification of HireRight service.
- Implements one of the following:
 - Transforms inbound HireRight alerts into HCM REST API calls to update verification and monitoring status related to certifications and licenses.
 - Transforms inbound HireRight alerts into HCM Data Loader DAT file format and loads into Oracle HCM Cloud.

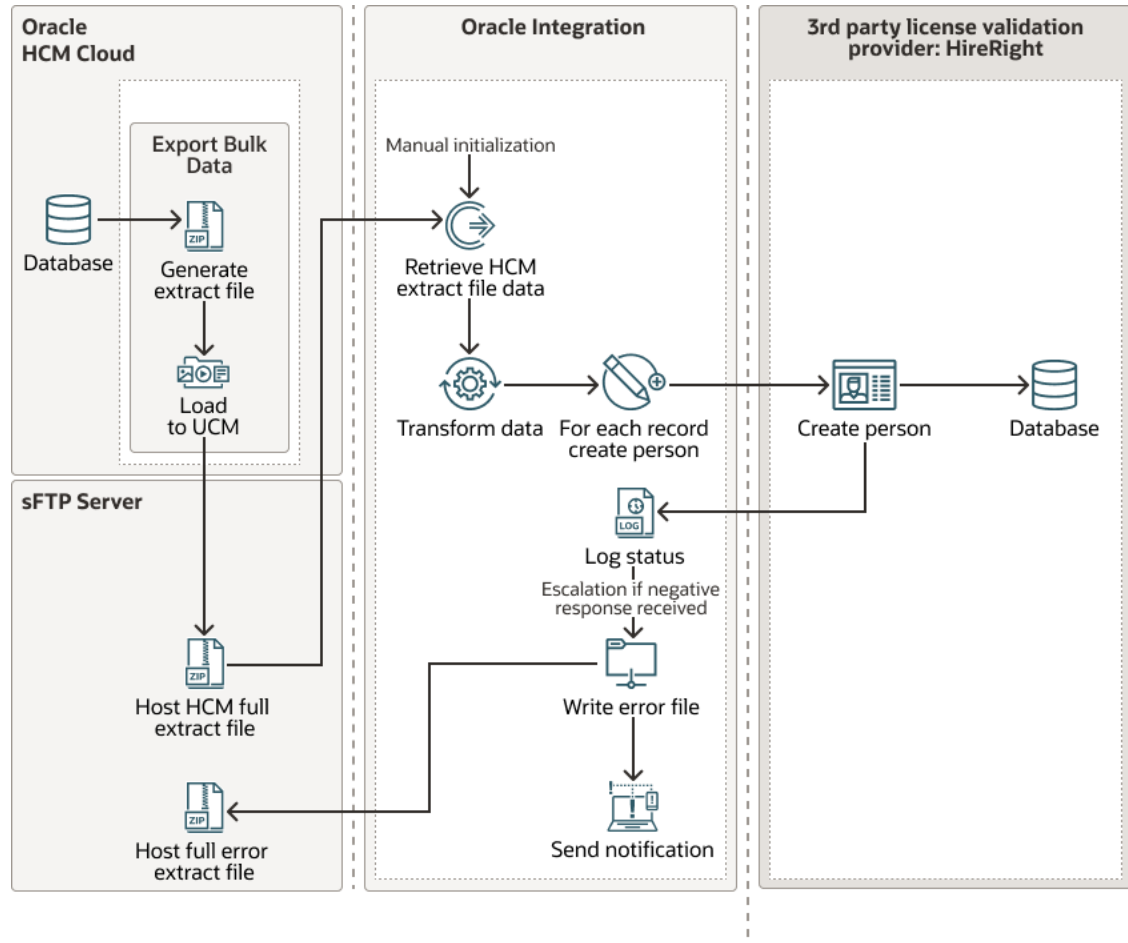
System and Access Requirements

- Oracle Integration. Version 24.06 or higher
- An account in Oracle Integration with the Service Invoker role
- An account in Oracle Integration with the Developer role
- Valid license for Oracle HCM Cloud version 24B or higher
- HireRight license and certifications verifications monitoring service agreement
- A secure FTP server (sFTP server) or File Server
- An FTP client to access the external sFTP server

Recipe Schema

This section provides an architectural overview of the integration flows in the recipe.

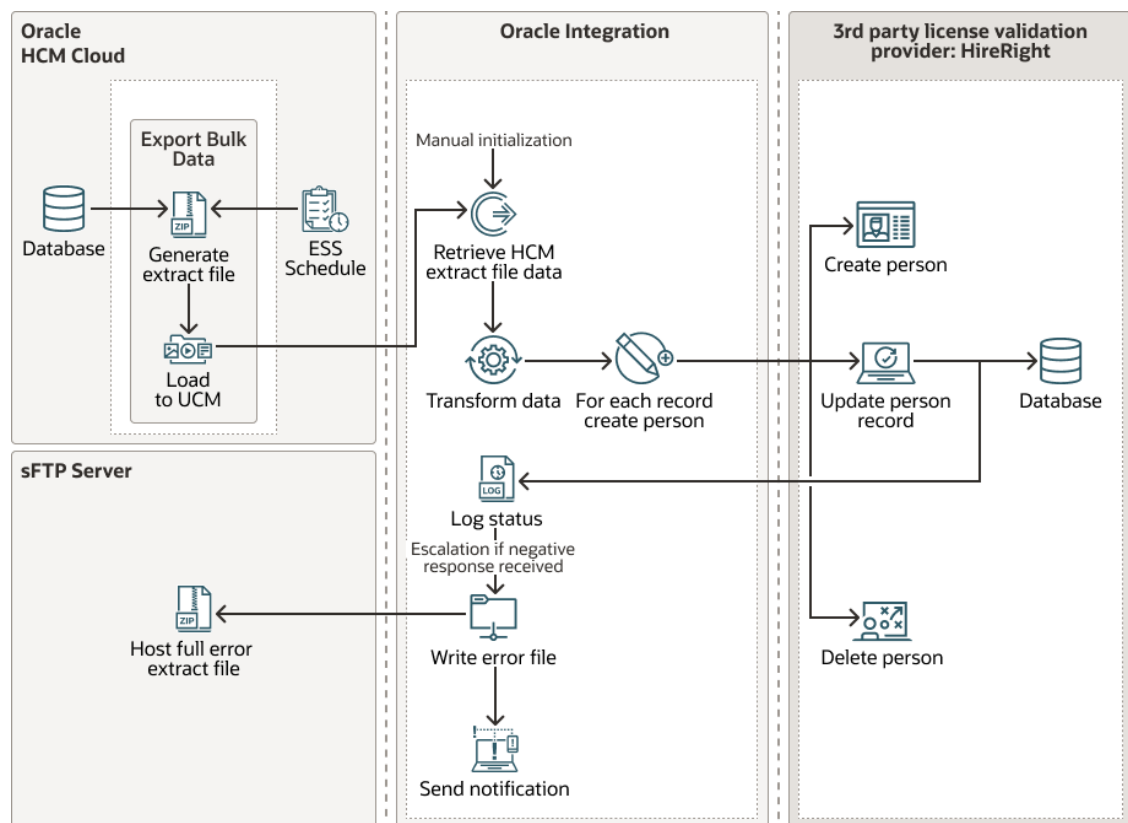
Oracle HCM Initial Outbound Integration to HireRight



This integration is used to create the initial HireRight roster of all employees within the Oracle HCM Cloud that need to be monitored. This integration should be run once.

A dedicated one-time HCM Extract should be run to generate a full file of all employees for inclusion on the HireRight roster. This file should be manually loaded on to an sFTP server which Oracle Integration is able to reach. The integration should be run after the first-time HCM extract file upload. The integration will retrieve the HCM Extract file from the sFTP server and for each employee record make a create person HireRight REST call. If errors are generated due to incomplete or incorrect data, an error file will be generated which is then written back to the sFTP server for manual review. A notification is sent confirming the final status of the integration.

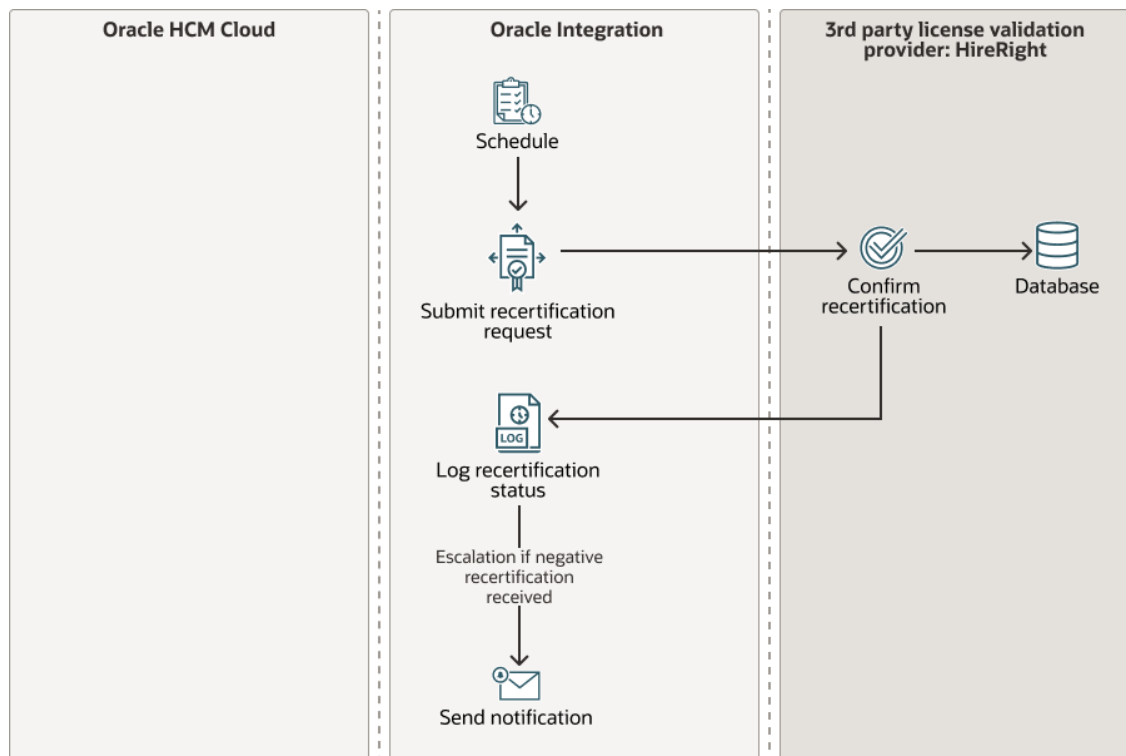
Oracle HCM Daily Outbound Integration to HireRight



This integration requires a daily HCM Extract to be run which generates a file containing only deltas from the previous run. This allows for changes entered in the HCM application to be sent to HireRight. Changes could include additions to the roster, deletions from the roster, status changes of existing employees, correction and updates to existing employees license and certificate information held by HireRight. This integration relies on the Oracle HCM Cloud Adapter to retrieve the generated HCM Extract file from UCM . This file is then processed by Oracle Integration into subsequent HireRight REST calls.

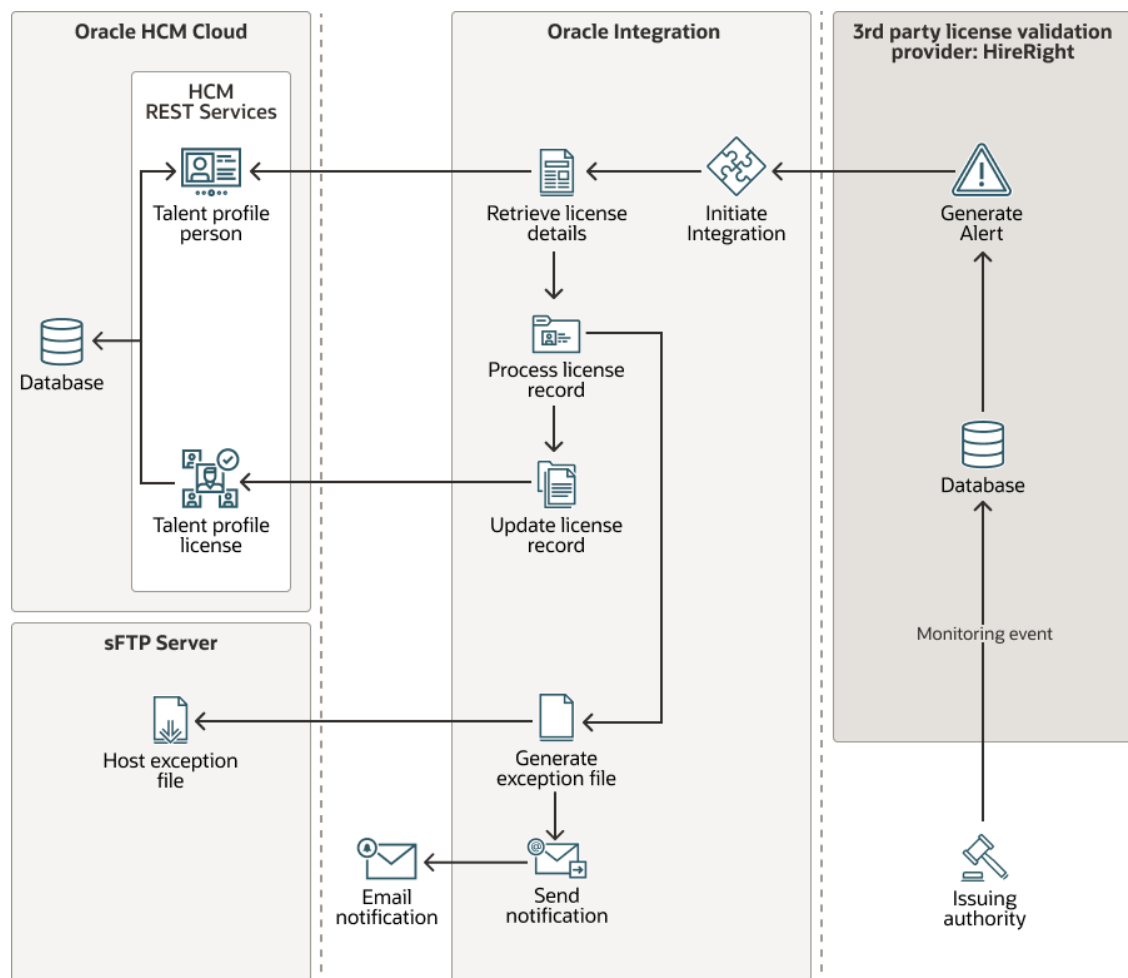
If errors are generated due to incomplete or incorrect data, an error file is created which is written to the sFTP server for manual review. A notification is also sent confirming the final status of the integration.

Oracle HCM Recertification Integration to HireRight



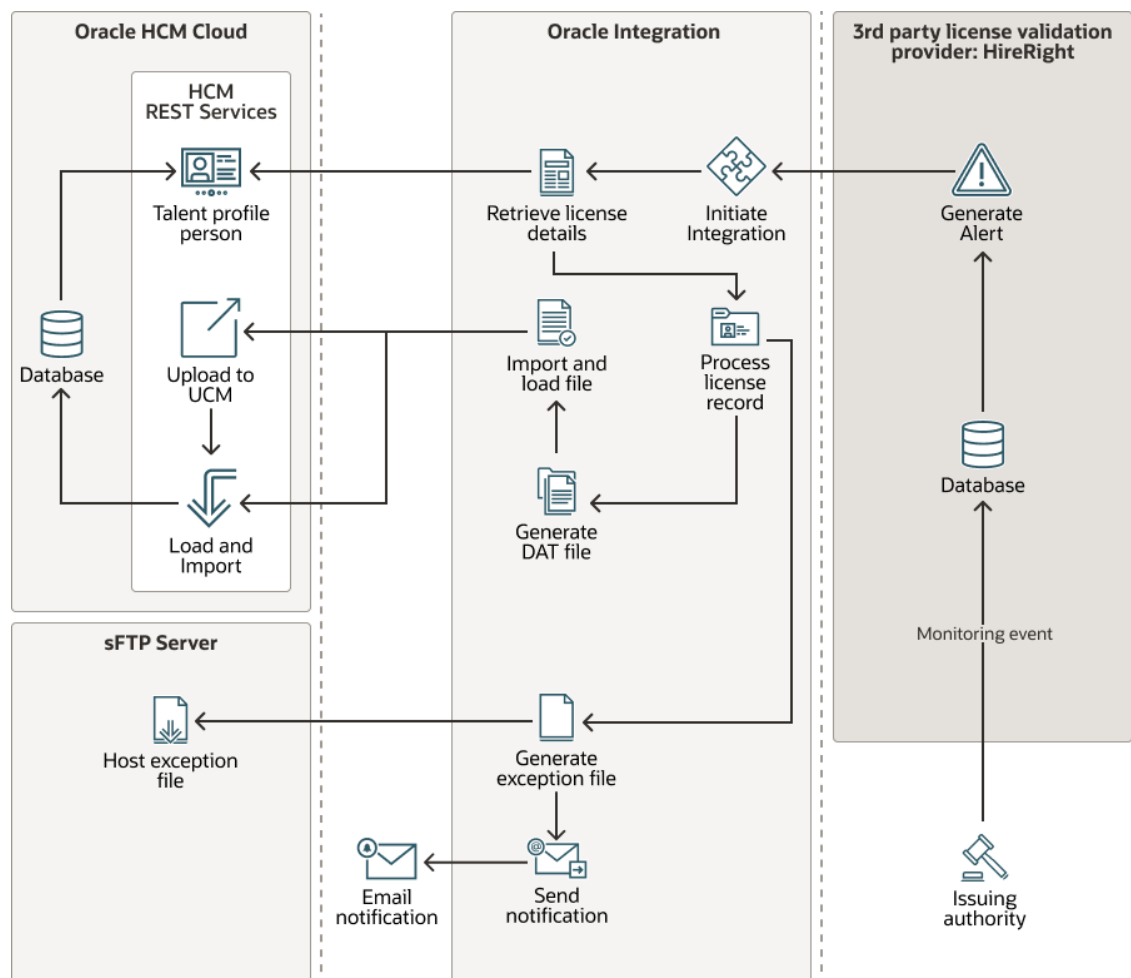
To facilitate monitoring compliance with federal and state law, HireRight automatically suspends monitoring program whenever a re-certification is required, meaning roster members are monitored, but any possible results are held until re-certification is received. This integration should be scheduled to run every day at the same time to ensure that re-certification with HireRight occurs. The integration makes a call to the HireRight certification endpoint and enacts re-certification. Any errors or issues result in a notification email to an Administrator who can take action to investigate further.

Oracle HCM Inbound Integration from HireRight (HCM REST)



HireRight generates alerts containing updates to license and certifications that are being monitored. These alerts are generated from various issuing authorities which are ultimately the primary source for licenses and certifications statuses. An individual alert is sent to Oracle Integration where it is transformed into an HCM REST API call and the corresponding individual Talent Profile Record of the employee is updated. As an optional step, a filter can be applied in which certain statuses are separated for manual review. You can also configure a lookup table to translate Issuing Authority statuses into seeded HCM Certification status. Both the filter and the lookup table are configurable. This exception file is in a DAT file format and can subsequently, be manually loaded into Oracle HCM Cloud via the HCM Data Loader.

Oracle HCM Inbound Integration from HireRight (HCM Data Loader)



When HireRight monitoring reveals new data related to the employees being managed, HireRight will push the data to Oracle Integration, which will cause this integration to be initiated. The integration transforms the data received from HireRight and makes additional API calls to identify the specific employee record and license or certificate being monitored. Oracle Integration then generates a DAT file and initiates an HDL Data Load and Import process to update the data into Oracle HCM Cloud. As an optional step, a filter can be applied in which certain statuses are separated for manual review. You can also configure a lookup table to translate Issuing Authority statuses into seeded HCM Certification status. Both the filter and the lookup table are configurable.

Assumptions

- Based on your requirements, you may enhance the recipe to validate or selectively filter employees which should not be created in HireRight.
- The integration flows of the recipe include the minimal data mapping required to synchronize the data with HireRight. You can have additional mappings based on your specific requirements.
- The integration flows of the recipe include sample error handling and notifications. A robust error handling based on your own environment can be additionally implemented.
- You must configure only one of the Oracle HCM Inbound integrations from HireRight, either the one using HCM DataLoader or the one using HCM REST APIs. Using both will result in errors and may impact business operations.

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud instance, HireRight instance and sFTP server in order to successfully connect to these external systems using Oracle Integration and import verified employees.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration, you'll require a separate user account on Oracle HCM Cloud if you do not already have one in place.

Log in to the Oracle HCM Cloud instance as an **Administrator** and perform the following tasks:

1. Create a user account for Oracle Integration which will be required to connect to Oracle HCM Cloud from Oracle Integration.
2. Assign roles to the user account. See *Assign Required Roles to an Integration User in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

For more information, see *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure HireRight

To access HireRight from Oracle Integration, you must perform certain configurations on your HireRight instance.

Note that the following tasks will involve the coordination with a HireRight implementation specialist.

1. API endpoints.
 - Production: Provided by your HireRight implementation specialist.
 - Staging: Provided by your HireRight implementation specialist.
2. API Header information

These values are required for every integration against HireRight.

Header	Description
hr-API-key	The API key is unique per client. HireRight will grant an API key for each client. If client uses REST connect API for background checks ordering, hr-API-key will be the same for background checks and monitoring roster management API calls.
hr-companylogin	Company login in HireRight system. String, max length: 7
hr-user-ref-id	Reference of the user (UserRefId) in HireRight system String, max length: 50
hr-APIversion	Version of Monitoring Roster and Alerts API. Current version is v1. String, max length: 10

3. Setup the following flex field in HireRight.

- clientPersonId

To know more about how to set up this flex field, contact your HireRight implementation specialist.

4. Retrieve the monitoring values.

These values are required for several of the integrations against HireRight where a payload is sent.

Value	Description
hr_MonitoringId	The id of the service that you subscribe to with HireRight.
hr_Monitoringname	The name of the service that you subscribe to with HireRight.

In addition to the above tasks, to access Oracle Integration from HireRight when sending alerts, Oracle recommends that you create a user specifically for HireRight. To complete this task, see Manage Access and Assign Roles in *Provisioning and Administering Oracle Integration 3*. Once the user account has been created, share this information with your HireRight implementation specialist. You'll also have to share the endpoint URL for the Oracle HCM Inbound Integration from HireRight using either HCM DataLoader or REST API.

Access an sFTP Server

You have to access an sFTP server and create necessary folders in it to host the HCM Extract for the initial load as well as any error files that are generated.

You can also use the embedded sFTP server (File Server) in Oracle Integration. To use File Server, you must enable it for your Oracle Integration instance. See *Enable File Server in Using File Server in Oracle Integration 3*.

Note that if you set up File Server, you may need to make modifications to the integrations in the recipe based on the implementation pattern that your Oracle Integration administrator has set up for your File Server.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

- [Configure the Oracle HCM Cloud Connection](#)
- [Configure the Oracle HCM Cloud REST Connection](#)
- [Configure the HireRight REST Connection](#)
- [Configure the sFTP Connection](#)
- [Configure the OIC REST Connection](#)
- [Configure the Lookup Table](#)
- [Configure the Oracle HCM Initial Outbound Integration to HireRight](#)
- [Configure the Oracle HCM Daily Outbound Integration to HireRight](#)
- [Configure the Oracle HCM Recertification Integration to HireRight](#)

- [Configure the Oracle HCM Inbound Integration from HireRight \(HCM REST\)](#)
- [Configure the Oracle HCM Inbound Integration from HireRight \(HCM Data Loader\)](#)

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle HCM Cloud REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the Oracle HCM Cloud REST endpoint. For example: <code>https://servername.fa.us2.oraclecloud.com/hcmRESTAPI/resources/latest/</code>

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the HireRight REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the port number of HireRight's sFTP server.

3. In the Security section, select **No Security Policy**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

For further information on your HireRight API configuration, contact your HireRight implementation specialist.

Configure the sFTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter the port number of your sFTP server.
SFTP Connection	Select Yes .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the OIC REST Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the **Security Policy** as **Basic Authentication**.
3. In the Access Type section, ensure **Public gateway** is selected.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

This recipe includes a lookup table called **HireRightStatusLookup** which is used to translate license statuses that HireRight captures from the issuing authorities and sends to the Oracle HCM Inbound integration, to setup statuses in your HCM Cloud instance. Based on the nature of your business, the values can be different, so you can configure this lookup table according to your needs.

1. In the Lookups section, click the lookup name.
2. If required, update and map the values in the **hireRightLicenseStatus** column to corresponding values in the **hcmLicenseStatus** column.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Oracle HCM Initial Outbound Integration to HireRight

You must configure the Oracle HCM Initial Outbound Integration to HireRight (**ORCL_HCM_LCNSVLDTN_HRRGHT_INITIAL_ROSTER**) integration flow to successfully extract the existing employee details from Oracle HCM Cloud along with relevant licenses and certifications required for monitoring. Oracle Integration will then generate a file which can be used to create corresponding employees and certifications in HireRight.

You can extract the following employee information:

- Basic details of the employee such as Person Number, First Name, Last Name, Person Email Address, Phone Number, and Home Address.
- Any licenses and certifications associated to the *job* assigned to the employee.
- Any licenses and certifications associated to the *position* assigned to the employee.

The following high-level steps are required to configure the Oracle HCM Initial Outbound Integration to HireRight integration flow:

1. [Configure the Oracle HCM Cloud Instance Extract](#)
2. [Submit an Extract](#)
3. [Update Integration Properties](#)
4. [Update Schedule Parameters](#)

Configure the Oracle HCM Cloud Instance Extract

Upload the pre-built extract to the Oracle HCM Cloud instance, update it (if required) and configure it.

1. Log in to the Oracle HCM Cloud instance.
2. Navigate to **Manage HCM Extract Definitions**.
From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Extract Definitions** or **Setup and Maintenance**, and then select **Manage HCM Extract Definitions**.
3. On the Extract Definitions page, click **Import**.
4. Enter the extract name and upload the extract file.
5. Click **OK**, then click **OK** again.
6. Next click **Edit** to the newly created extracted.
7. Select **Extract Execution Tree** and click **Compile All Formula**.
8. Refresh the screen and make sure all the fast formula status show passed status.
9. Click **Export XML Schema** to download the .xsd file.
10. Click **Save and Close**.

Submit an Extract

You have to submit the HCM Extract in order to configure the Oracle HCM Initial Outbound Integration to HireRight integration flow in Oracle Integration. The integration will use this submitted extraction for fetching data while running.

1. From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Submit Extract**.
2. Search for the extract with its name.
3. Click the extract.
4. Enter the extract instance name and select the effective date.
5. Click **Submit**.

This will execute the one-time HCM Extract of all valid employees for monitoring

Update Integration Properties

You can update the integration properties with appropriate values based on your environment and business rules.

1. In the Integrations section, click **Actions** ... on the **ORCL_HCM_LCNSVLDTN_HRRGHT_INITIAL_ROSTER** integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo:** This integration property holds the email address of the individual or distribution group to which error notifications are to be sent.
 - **RetryCount:** This integration property holds the number of retries for an individual record to be written to HireRight before an error is declared and written to the exception log. The default value is 2.

3. Click **Submit**.
A message confirms that the integration properties have been updated.

Update Schedule Parameters

You can update the following parameters with appropriate values based on your environment and business rules.

1. In the Integrations section, click the **ORCL_HCM_LCNSVLDTN_HRRGHT_INITIAL_ROSTER** integration flow to open it.
2. On the integration canvas, click the **Schedule** action and select **Edit**.
3. Update the following parameters with appropriate values.

Field	Information to Enter
hr-API-key	Enter the value for this field. See Configure HireRight .
hr-company-login	Enter the value for this field.
hr-user-ref-id	Enter the value for this field.
hr-API-version	Enter the value for this field
hr_MonitoringId	Enter the value for this field
hr_Monitoringname	Enter the value for this field
schedule_param_ErrorFilePath	Specify the path of the folder on your sFTP server where exception records are written for further processing and manual review.
schedule_param_FileName	Specify the name of the HCM Extract result file that was generated earlier.
schedule_param_FilePath	Specify the path of the folder on your sFTP server where the HCM Extract result file is located.

4. Click **Save**.

Leave the rest of the values in the Schedule action as is since they're referenced throughout the recipe.

Configure the Oracle HCM Daily Outbound Integration to HireRight

You must configure the Oracle HCM Daily Outbound Integration to HireRight (**ORCL_HCM_LCNSVLDTN_HRRGHT_DAILY_ROSTER**) integration flow to run the HCM Extract daily so as to generate a file containing only deltas from the previous run.

You can extract the following employee information:

- Basic details of the employee such as Person Number, First Name, Last Name, Person Email Address, Phone Number, and Home Address.
- Any licenses and certifications associated to the *job* assigned to the employee.
- Any licenses and certifications associated to the *position* assigned to the employee.

The following high-level steps are required to configure the Oracle HCM Initial Outbound Integration to HireRight integration flow:

1. [Configure the Oracle HCM Cloud Instance Extract](#)
2. [Submit an Extract](#)
3. [Schedule Extract Submission](#)
4. [Update Integration Properties](#)

5. [Update Schedule Parameters](#)
6. [Change the Extract Configuration](#)

Configure the Oracle HCM Cloud Instance Extract

1. Log in to the Oracle HCM Cloud instance.
2. Navigate to **Manage HCM Extract Definitions**.
From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Extract Definitions** or **Setup and Maintenance**, and then select **Manage HCM Extract Definitions**.
3. On the Extract Definitions page, click **Import**.
4. Enter the extract name and upload the extract file.
5. Click **OK**, then click **OK** again.
6. Next click **Edit** to the newly created extracted.
7. Select **Extract Execution Tree** and click **Compile All Formula**.
8. Refresh the screen and make sure all the fast formula status show passed status.
9. Click **Export XML Schema** to download the .xsd file.
10. Click **Save and Close**.

Note

The HCM Extract used for the daily integration is different than the Initial Extract definition used in the previous integration. The daily integration will fail if you use the Initial Extract definition.

Submit an Extract

You have to submit the HCM Extract in order to configure the Oracle HCM Daily Outbound Integration to HireRight integration flow in Oracle Integration. The integration will use this submitted extraction for fetching data while running.

1. From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Submit Extract**.
2. Search for the extract with its name.
3. Click the extract.
4. Enter the extract instance name and select the effective date.
5. Click **Submit**.

This will execute the one-time HCM Extract of all valid employees for monitoring

Schedule Extract Submission

You must schedule an extract submission to fetch the new data per the required frequency for running the Oracle HCM Daily Outbound Integration to HireRight integration flow in Oracle Integration.

1. From the **Fusion Navigation** menu, and then from **My Client Groups**, select the **Data Exchange** application. Under HCM Extracts, select **Submit Extract**.
2. Search for the extract with its name.

3. Click the extract.
4. Enter the extract instance name and select the effective date.
5. Select the required frequency for extract submission, and select the start and end date.

Note

Although you can set any value as minimum frequency, the best practice is to use the extract submission frequency as greater than or equal to daily. The recommended value is daily. This should be agreed with your EverCheck implementation specialist.

Update Integration Properties

You can update the integration properties with appropriate values based on your environment and business rules.

1. In the Integrations section, click **Actions** ... on the **ORCL_HCM_LCNSVLDTN_HRRGHT_DAILY_ROSTER** integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo:** This integration property holds the email address of the individual or distribution group to which error notifications are to be sent.
 - **RetryCount:** This integration property holds the number of retries for an individual record to be written to HireRight before an error is declared and written to the exception log. The default value is 2.
3. Click **Submit**.
A message confirms that the integration properties have been updated.

Update Schedule Parameters

You can update the following parameters with appropriate values based on your environment and business rules.

1. In the Integrations section, click the **ORCL_HCM_LCNSVLDTN_HRRGHT_DAILY_ROSTER** integration flow to open it.
2. On the integration canvas, click the **Schedule** action and select **Edit**.
3. Update the following parameters with appropriate values.

Field	Information to Enter
hr-API-key	Enter the value for this field. See Configure HireRight .
hr-company-login	Enter the value for this field.
hr-user-ref-id	Enter the value for this field.
hr-API-version	Enter the value for this field
hr_MonitoringId	Enter the value for this field
hr_Monitoringname	Enter the value for this field
lastDocumentID	The default value is 0. Enter a value, otherwise it gets automatically updated to reflect the value in UCM in the first run of the integration.

4. Click **Save**.

Leave the rest of the values in the Schedule action as is since they're referenced throughout the recipe.

Change the Extract Configuration

To ensure that the appropriate HCM extract is linked to this recipe, perform the following:

1. In the Integrations section, click the **ORCL_HCM_LCNSVLDTN_HRRGHT_DAILY_ROSTER** integration flow to open it.
2. On the integration canvas, click the Invoke element **HCMExtract** and select **Edit** from the menu.
3. Click **Operations** and update the integration name. The integration name should be same as the name added in the HCM Extract while configuring the Oracle HCM Cloud instance extract in Oracle HCM Cloud.
4. Click **Done**.
5. Save the integration.

Configure the Oracle HCM Recertification Integration to HireRight

To facilitate monitoring compliance with federal and state law, HireRight automatically suspends monitoring program whenever a re-certification is required, meaning roster members are monitored, but any possible results are held until re-certification is received. The Oracle HCM Recertification Integration to HireRight (**ORCL_HCM_LCNSVLDTN_HRRGHT_DAILY_RECERT**) integration flow should be scheduled to run every day at the same time to ensure that re-certification with HireRight occurs. The integration makes a call the HireRight certification endpoint and enacts re-certification. Any errors or issues will result in a notification email to an Administrator who can take action to investigate further.

The following high-level steps are required to configure the Oracle HCM Recertification Integration to HireRight integration flow:

1. [Update Schedule Parameters](#)
2. [Update Integration Properties](#)

Update Schedule Parameters

You can update the schedule parameters of the integration with appropriate values based on your environment and business rules.

1. In the Integrations section, click the **ORCL_HCM_LCNSVLDTN_HRRGHT_DAILY_RECERT** integration flow to open it.
2. On the integration canvas, click the **Schedule** action and select **Edit**.
3. Update the following parameters with appropriate values.

Field	Information to Enter
hr-API-key	Enter the value for this field.
hr-company-login	Enter the value for this field.
hr-user-ref-id	Enter the value for this field.
hr-API-version	Enter the value for this field.

4. Click **Save**.

Update Integration Properties

You can update the integration properties with appropriate values based on your environment and business rules.

1. In the Integrations section, click **Actions** ... on the **ORCL_HCM_LCNSVLDTN_HRRGHT_DAILY_RECERT** integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo:** This integration property holds the email address of the individual or distribution group to which error notifications are to be sent.
 - **RetryCount:** This integration property holds the number of retries for an individual record to be written to HireRight before an error is declared and written to the exception log. The default value is 2.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Configure the Oracle HCM Inbound Integration from HireRight (HCM REST)

Configure the Oracle HCM Inbound Integration from HireRight - HCM REST (**ORCL_HRRGHT_ALRT_INBOUND_REST**) integration flow to load the monitored and verified employee records received from HireRight to Oracle HCM Cloud.

The following high-level steps are required to configure the Oracle HCM Inbound Integration from HireRight (HCM REST) integration flow:

1. [Update Integration Properties](#)
2. [Configure the Integration Variables](#)
3. [Configure the Use of the Lookup Table in the Integration](#)

Update Integration Properties

You can update the integration properties with appropriate values based on your environment and business rules.

1. In the Integrations section, click **Actions** ... on the **ORCL_HRRGHT_ALRT_INBOUND_REST** integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo:** This integration property holds the email address of the individual or distribution group to which error notifications are to be sent.
 - **RetryCount:** This integration property holds the number of retries for an individual record to be written to HireRight before an error is declared and written to the exception log. The default value is 2.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Configure the Integration Variables

1. In the Integration section of the project workspace in Oracle Integration, click the **ORCL_HRRGHT_ALRT_INBOUND_REST** integration.
2. Click the **AssignGlobalParams** Assign action, and configure the following:

Field	Information to Enter
schedule_param_ErrorFilePath	Specify the folder location on your sFTP server where the file with exception records can be placed for further processing or manual review. These records are returned from HireRight, but are not automatically processed and loaded into Oracle HCM Cloud. They need to be validated.

Leave the rest of the values in the Assign action as is since they're referenced throughout the recipe.

Configure the Use of the Lookup Table in the Integration

Within the integration as part of the generation of a REST call to HCM, the existing values received from HireRight are mapped to Oracle HCM Cloud values. This is done using the lookup table configured in the recipe. See [Configure the Lookup Table](#).

The mapping occurs within the **Map to updateCertificate** element.

1. In the Integration section of the project workspace in Oracle Integration, click the Oracle HCM Inbound Integration from HireRight (HCM REST) integration.
2. Click the **Map to updateCertificate** element, and then click **Actions** and select **Edit** to open the mapper.
3. On the target side of the mapping screen, expand the **updateCertificateRequest (REST) Record Set Name** until the **Status** field is shown.
4. Select the **Status** field to see the expression for the field.
At this point, you can do one of the following, depending on your requirements:
 - Leave as is and leverage the default values of the lookup table within the integration to transform the inbound Status to reflect the values within Oracle HCM Cloud. This may require additional values to be added to the lookup table and configurations in Oracle HCM Cloud.
 - Remove the lookup table reference and simply map the values which are being returned from HireRight to the Statuses maintained in Oracle HCM Cloud. To achieve this, clear the Expression box and drag the Status on the source side to the Status on the target side. Note that additional configurations may also need to be made in Oracle HCM Cloud.

Configure the Oracle HCM Inbound Integration from HireRight (HCM Data Loader)

Alternatively, configure the Oracle HCM Inbound Integration from HireRight - HCM Data Loader (**ORCL_HRRGHT_ALRT_INBOUND_HDL**) integration flow and use the HCM Data Loader (HDL) to load the monitored and verified employee records received from HireRight to Oracle HCM Cloud.

The following high-level steps are required to configure the Oracle HCM Inbound Integration from HireRight (HDL) integration flow:

1. [Update Integration Properties](#)

2. [Configure the Use of the Lookup Table in the Integration](#)

Update Integration Properties

You can update the integration properties with appropriate values based on your environment and business rules.

1. In the Integrations section, click **Actions** . . . on the **ORCL_HRRGHT_ALERT_INBOUND_HDL** integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration property with appropriate values.
 - **EmailTo:** This integration property holds the email address of the individual or distribution group to which error notifications are to be sent.
3. Click **Submit**.
A message confirms that the integration property has been updated successfully.

Configure the Use of the Lookup Table in the Integration

Within the integration as part of the generation of a HDL file, the existing values received from HireRight are mapped to Oracle HCM Cloud values. This is done using the lookup table configured in the recipe. See [Configure the Lookup Table](#).

The mapping occurs within the **Map to writeDATFile** element.

1. In the Integration section of the project workspace in Oracle Integration, click the Oracle HCM Inbound Integration from HireRight (HDL) integration.
2. Click the **Map to writeDATFile** element, and then click **Actions** . . . and select **Edit** to open the mapper.
3. On the target side of the mapping screen, expand the **writeDATFile Response (Stage File) Record Set Name** until the **Status** field is shown.
4. Select the **Status** field to see the expression for the field.
At this point, you can do one of the following, depending on your requirements:
 - Leave as is and leverage the default values of the lookup table within the integration to transform the inbound Status to reflect the values within Oracle HCM Cloud. This may require additional values to be added to the lookup table and configurations in Oracle HCM Cloud.
 - Remove the lookup table reference and simply map the values which are being returned from HireRight to the Statuses maintained in Oracle HCM Cloud. To achieve this, clear the Expression box and drag the Status on the source side to the Status on the target side. Note that additional configurations may also need to be made in Oracle HCM Cloud.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** . . . on each of the integration flows, then select **Run**.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule the integrations to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In your Oracle HCM Cloud instance, check if the verified employee licenses and certificates have been imported.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*
- [Getting Started with HCM Extract](#)
- [Creating and Loading Files with HCM Data Loader \(HDL\)](#)
- [Load data into Oracle Fusion Cloud Human Capital Management \(HCM\)](#)
- [REST API for Oracle Fusion Cloud HCM](#)

Generate an ANSI 834 Benefits File from Oracle HCM Cloud Extract File

Use this recipe to generate an ANSI 834 benefits file from an Oracle HCM Cloud data extract file.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how you can use the data mapper capabilities in Oracle Integration to transform standard data extracted from Oracle HCM Cloud into the data structure of an ANSI 834 benefits provider file.

To use the recipe, you must install the recipe and configure the connection and other resources within it. Subsequently, you can activate the integration flow of the recipe. When triggered, the integration flow of the recipe performs the following:

- Reads the Oracle HCM Cloud data extract file from an FTP server.
- Transforms and formats the extracted data using the Oracle Integration data mapper.
- Generates a valid ANSI 834 benefits provider file.

- Uploads and writes the ANSI 834 benefits provider file to the FTP server.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- A secure FTP (SFTP) server
- An FTP client to access the SFTP server
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud instance and SFTP server in order to successfully connect to these external systems using Oracle Integration and generate ANSI 834 benefits file from Oracle HCM Cloud extracted data file.

1. Ensure that data is extracted from Oracle HCM Cloud.

Use the Oracle HCM Cloud adapter to retrieve files from an Oracle HCM Cloud instance via the HCM Extract tool. To access the Oracle HCM Cloud instance from Oracle Integration, see Prerequisites for Creating a Connection in *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Note that the file with the extracted data has to be named as **SubscribersHcmExtract.xml** for the integration flow of the recipe to process the file. Put this file in the FTP server's input folder. You can download the sample SubscribersHcmExtract.xml file to use in the recipe from [Sample file](#).

2. Access your FTP server.
 - a. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
 - b. Create an output folder. This is the folder where the ANSI 834 benefits output files will be uploaded.
 - c. Create an input folder. This is the folder from where the Oracle HCM Cloud extracted data input files will be read.

For the purpose of testing, you can download a sample input file and place it in the input folder. For more information, see [Related documents and sample files](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the following integration properties with appropriate values.
 - a. **InputFolder**: This integration property holds the path of the input folder that you created in your FTP server. See [Before You Install the Recipe](#).
 - b. **OutputFolder**: This integration property holds the path of the output folder that you created in your FTP server.
 - c. **FromAddress**: This integration property holds the email address from which error notification emails are sent. Enter an email address of your choice.
 - d. **ToAddress**: This integration property holds the email address to which error notification emails are sent. Enter an email address of your choice.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if an ANSI 834 benefits file has been uploaded to the output folder of your FTP server.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Generate and Email Patient Report Summary Using OCI AI Services

Use this recipe to analyze and extract relevant information from patient reports, summarize reports, and send email notifications with report summaries that include diagnosis, treatment plans and recommended next steps using the artificial intelligence capabilities of the following OCI AI services - Oracle Cloud Infrastructure Vision (OCI Vision) and Oracle Cloud Infrastructure Generative AI (OCI Generative AI).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe uses OCI Vision to analyze and extract relevant information from a patient report, then leverages OCI Generative AI capabilities to summarize the report, and finally sends email notification to the patient with the report summary.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When the main integration flow of the recipe receives the patient report, it uses OCI Vision to analyze and extract relevant information from the report including the patient ID. Then it invokes the child integration that uses OCI Generative AI to generate a summary of the report including diagnosis, treatment plans, recommended next-steps and so on. The recipe uses the patient ID from the report to retrieve required details about the patient such as the patient's email address from Oracle Health. Finally, it sends an email notification to the patient with the report summary.

System and Access Requirements

- Oracle Integration 3
- OCI Vision

- OCI Generative AI
- Oracle Health

Before You Install the Recipe

To use the OCI AI services - OCI Vision and OCI Generative AI, and to successfully generate patient report summary you have to complete certain prerequisites in Oracle Cloud Infrastructure Console.

Complete the tasks described in Prerequisites for OCI AI services in *Using Integrations in Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Health Connection

Note

To get patient details from Oracle Health, you can use the open sandbox instance that allows developers to experiment with the service without requiring authentication.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the service root URL of the Oracle Health open sandbox instance. See [Service Root URL](#) in *FHIR R4 APIs for Oracle Health Millennium Platform*.
3. In the Security section, select **No Security Policy**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Integration Flows

Edit the integration flows to configure actions for the following OCI AI services.

- [OCI Vision](#)
- [OCI Generative AI](#)

OCI Vision

1. In the Integrations section, click the **Summarize Health Report - AI** integration flow.
2. On the integration canvas, click the **OCI Vision** element, and then click **Actions** ... and select **Edit**.
3. Click **Next** and select the OCI compartment.
4. Click **Next** and then **Done**.
5. Click **Save** to save the integration.
6. To return to the project workspace, click **Go back** .

To know more about the vision action, see Analyze and Extract Information from Images with a Vision Action in *Using Integrations in Oracle Integration 3*.

OCI Generative AI

1. In the Integrations section, click the **Generate Text using GenAI** integration flow.
2. On the integration canvas, click the **OCI Generative AI** element, and then click **Actions** ... and select **Edit**.
3. Click **Next** and select the OCI region, OCI compartment, model, and model ID.
4. Click **Next** and then **Done**.
5. Click **Save** to save the integration.
6. To return to the project workspace, click **Go back** .

To know more about the generative AI action, see Generate Text with a Generative AI Action in *Using Integrations in Oracle Integration 3*.

Configure the Lookup Table

Edit the lookup table **CommonLookup** and enter values for the lookup keys as per your requirement.

1. In the Lookups section, click the lookup name.
2. Update the lookup keys with appropriate values

Key	Description
CompartmentId	Specify the OCI compartment for OCI Generative AI.

Key	Description
ModelId	Specify the AI model to use for summarizing the patient report. For example: <code>cohere.command-a-03-2025</code> For details about pretrained generative AI models, see Chat Models in Oracle Cloud Infrastructure documentation.
ServingType	Specify the serving type for the model. For example: <code>ON_DEMAND</code>
MaxTokens	Specify the maximum number of tokens that you want OCI Generative AI to generate. For example: <code>600</code>
Temperature	This is a tuning parameter for AI response. Adjust for either safe, predictable text (low) or creative, diverse text (high).
FrequencyPenalty	This is a tuning parameter for AI response. A high value generates less repetition.
PresencePenalty	This is a tuning parameter for AI response. A high value encourages new ideas or words to appear in the text.
TopP	This is a tuning parameter for AI response. Controls randomness and coherence based on cumulative probability.
TopK	This is a tuning parameter for AI response. Controls randomness and coherence based on a fixed count.
ApiFormat	Specify API response format.
TempEmailId	Email address of the recipient to whom the email notification with the patient report summary have to be sent.

- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Test the recipe in Oracle Integration.
 - In the Integrations section of the project workspace, click **Actions**  on the **Summarize Health Report - AI** integration flow, then select **Run**.
 - On the Configure and run page, in the Request section, click **File**.
 - Upload a valid patient report. Note that the uploaded file must be in image format (JPG, PNG).
 - Click **Run**
- Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
- Check if an email is received with the patient report summary in the patient's email address.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)
- [OCI Vision](#)
- [OCI Generative AI](#)

Generate Oracle Hospitality Access Tokens

Use this recipe to generate an access token from the Oracle Hospitality Integration Platform (OHIP).

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 and configure the connection and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow 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 and Access 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*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Hospitality Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the Gateway URL obtained from the Oracle Hospitality Developer Portal. See Assumptions .

3. 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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table to enter the necessary details.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

Key	Description
X-App-Key	Enter the Application Key obtained from the Oracle Hospitality Developer Portal. See Assumptions .
ToEmailID	Enter the email address to which run-time exception emails are to be sent.
FromEmailID	Enter the email address from which run-time exception emails are to be sent.

3. Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration property values.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.
 - **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.
 - c. Click **Submit**.

A confirmation message is displayed informing that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**. Note that the **Ad hoc request** radio button is selected by default.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)

Get Details of a Facebook Page

Use this recipe to retrieve the details of a Facebook page.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)

- [Activate and Run the Recipe](#)

About This Recipe

This recipe can perform the following:

- 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 use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe contains two integration flows: **Oracle Facebook Get Page Posts** and **Oracle Facebook Get Page Feed**.

- To get all posts published by a Facebook page, trigger the **Oracle Facebook Get Page Posts** integration flow by providing the page-id of the particular Facebook page in a REST request. The response returns all posts published by the Facebook page.
- To get entire feed of a Facebook page, trigger the **Oracle Facebook Get Page Feed** integration flow by providing the page-id of the particular Facebook page in a REST request. The response returns entire feed of the Facebook page.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Facebook
- An account on Facebook with the Administrator role

Before You Install the Recipe

To access Facebook using Oracle Integration and retrieve details of a Facebook page, you must perform certain configuration tasks on your Facebook instance.

For detail information, see Prerequisites for Creating a Connection in *Using the Facebook Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Facebook Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Facebook OAuth Authorization Code Credentials .

Field	Information to Enter
Client ID	Enter the client ID. See Prerequisites for Creating a Connection in <i>Using the Facebook Adapter with Oracle Integration 3</i> .
Password	Enter the client secret.
Scope	Enter the following scopes, each separated by a blank space. user_posts user_likes publish_actions manage_pages

- Click **Provide Consent** to allow consent.
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the REST Connection

- In the Connections section, click the connection name.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the **Oracle Facebook Get Page Posts** integration.
 - In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - On the Configure and run page, in the Request section, enter the page-id of the Facebook page.
 - Click **Run**.
 - Monitor the running of the integration flow in Oracle Integration.
 - In the project workspace, click **Observe**. You'll see the integration flow being triggered and running successfully.
 - To manage errors in your project, see [Manage Errors in a Project](#).
 - In the response, all the posts published to the Facebook page are returned.

3. Run the **Oracle Facebook Get Page Feed** integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, enter the page-id of the Facebook page.
 - c. Click **Run**.
 - d. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
 - e. In the response, the entire feed of the Facebook page is returned.

Related Documentation

- *Using the Facebook Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Get Incident Details From Oracle B2C Service

Use this recipe to get incident details from Oracle B2C Service for an incident ID.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate the integration flow of the recipe and send a REST request with the incident ID to the integration's endpoint URL from an external application. When triggered, the recipe's integration flow fetches the incident details corresponding to the incident ID from Oracle B2C Service and returns the incident details as a response to the external application. The response payload is also logged in Oracle Integration, which you can view in the activity stream.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Oracle B2C Service
- An account on Oracle B2C Service with the Administrator role
- At least one incident created on your Oracle B2C Service instance

Before You Install the Recipe

To access Oracle B2C Service using Oracle Integration and fetch incident details for a given incident ID, you must log in to your Oracle B2C Service as an **Administrator** and complete

certain prerequisite tasks. See *Prerequisites for Creating a Connection in Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle B2C Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the WSDL to use. See *Prerequisites for Creating a Connection in Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle B2C Service.
Password	Enter the password of the account created for Oracle Integration on Oracle B2C Service.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured.

A message confirms if your test is successful.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.

- a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send the required REST command to this endpoint URL with the incident ID as the request parameter.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the incident ID as the request parameter.
 - c. Click **Run**.

The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success status 200 OK, and the response returned.
4. View the response message in the Activity stream pane to check if the incident details corresponding to the incident ID are returned.

Related Documentation

- *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Get Opportunity Details from Oracle B2B Service

Use this recipe to get opportunity details from Oracle B2B Service for an opportunity ID and send the opportunity details to the caller as a response.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to get opportunity details from Oracle B2B Service for an opportunity ID and send the opportunity details to the caller as a response.

The Oracle CX Sales and B2B Service Adapter is invoked in the integration, and is configured to get opportunity details from Oracle B2B Service. The opportunity 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 and Access Requirements

- Oracle B2B Service
- Oracle Integration 3

Before You Install the Recipe

To access the Oracle CX Sales and B2B Service instance from Oracle Integration, you'll require a separate user account on Oracle CX Sales and B2B Service.

Perform the following tasks to configure Oracle CX Sales and B2B Service for this recipe.

1. Create a new user account for Oracle Integration. You'll use the credentials of this user account to connect to Oracle CX Sales and B2B Service from Oracle Integration.

For the procedure to create the integration user account, see *Create an Integration User Account* in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.

2. Assign the following roles to the user account:

- Integration Specialist
- AttachmentsUser
- FND_MANAGE_CATALOG_SERVICE_PRIV

For steps to assign roles, see *Assign Integration Roles* in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.

3. Make a note of the user name and password you set for the account.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle CX Sales and B2B Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Connection

1. In the Connections section, click the connection name.
2. On the Oracle REST Connection page, click **Test** to ensure that your connection is correctly configured.

A message confirms if your test is successful.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the ID of the opportunity you want to retrieve from Oracle B2B Service. Enter the opportunity ID in the POST request's **URI Parameters** tab.

The recipe retrieves the corresponding opportunity details from Oracle B2B Service and sends the opportunity details to the caller as a response.

3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, enter the ID of the opportunity that you wish to retrieve.
 - c. Click **Run**.

The recipe invokes a REST API to get the opportunity details from Oracle B2B Service.
 - d. In the Response section of the test page, under the **Body** tab, you'll find a success response, 200 OK.

Related Documentation

- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*

- *Using the REST Adapter with Oracle Integration 3*

Get Organization Details from Oracle Service Cloud

Use this recipe to retrieve details of an organization from Oracle Service Cloud (RightNow).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how you can get details of an organization from Oracle Service Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe has two integration flows: **Oracle Service Cloud Create Org For Sales Acct** and **Oracle SOAP ServiceCloud Get Organization**.

- The **Oracle Service Cloud Create Org For Sales Acct** integration flow creates an organization in Oracle Service Cloud for a corresponding account created in Oracle Sales Cloud; and verifies the correlation.
- The **Oracle SOAP ServiceCloud Get Organization** integration flow is triggered when you provide the organization ID in a SOAP request. The details of the organization such as the address that includes street, city, postal code, and country is returned in the response.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462), or higher
- Oracle Service Cloud (RightNow)
- An account on Oracle Service Cloud with the Administrator role
- Oracle CX Sales and B2B Service
- An account on Oracle CX Sales and B2B Service with the Administrator role

Before You Install the Recipe

You must perform certain configuration tasks on your Oracle Service Cloud and Oracle CX Sales and B2B Service instances in order to successfully connect to these external systems using Oracle Integration and get organization details from Oracle Service Cloud for corresponding organizations created in Oracle CX Sales and B2B Service.

For more information about the configuration tasks that you have to perform in Oracle Service Cloud, see Prerequisites for Creating a Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.

For more information about the configuration tasks that you have to perform in Oracle CX Sales and B2B Service, see Prerequisites for Creating a Connection in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Engagement Cloud Instance Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle RightNow ORG Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the WSDL URL of the Oracle Service Cloud instance.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Service Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the WSDL URL of the Oracle Service Cloud instance.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration in the Oracle Service Cloud instance. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration in the Oracle Service Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle SOAP Get Organization Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the WSDL URL of the Oracle Service Cloud instance.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on the Oracle Service Cloud instance. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on the Oracle Service Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle SOAP ServiceCloud Get Organization** integration flow, then select **Run Details**.
 - b. In the panel that opens, copy the **Metadata URL**. This is the integration's endpoint URL.
 - c. From the external application such as SoapUI, load the WSDL hosted at the **Metadata URL** and send a request to the available operation.
 - d. Provide the Organization ID as argument for the ID field.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if the organization details are returned in the response.

Related Documentation

- *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
- *Using the SOAP Adapter with Oracle Integration 3*

Get Stocks Data

Use this recipe to get stock-related data of a company from an external source and send the details to the caller as a response.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to get 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 install the recipe and configure the connections within it. Subsequently, you can activate the integration in the recipe and send a REST request from an external application with the company symbol, date, and the API key for the stock application to the integration's endpoint URL. When triggered, the integration flow fetches the stock related information for the particular company on the specified date from the stock application, and returns this information in the response.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- An account on an application such as Polygon.io that provides stock market data via REST APIs

Before You Install the Recipe

You must perform the following configuration tasks on Polygon.io in order to connect to it using Oracle Integration, and retrieve stock-related information for a company on a specified date with its stock market data APIs.

To configure Polygon.io:

1. Log in to Polygon.io at <https://polygon.io/dashboard/login>.
2. Click **Dashboard** and select **API Keys**.
3. By default there is one API Key. You can generate a new key by clicking **Add API Key**.
4. Note down the value of the API Key.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Get Stock Quote Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the following URL: https://api.polygon.io/v1/open-close .

3. In the Security section, select **No Security Policy**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Sample Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a GET request to this endpoint URL along with the following request parameters:
 - **date**
 - **symbol**
 - **apiKey**
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the values for the following request parameters:
 - **date**: Specify the date in the format YYYY-MM-DD.
 - **symbol**: Specify the company's stock symbol. For example: ORCL
 - **apiKey**: Enter the value of the API key that you obtained in Polygon.io. See [Before You Install the Recipe](#).
 - c. Click **Run**
 - d. In the Response section of the Configure and run page, you'll find a success status, 200 OK, and the response returned.
4. View the request and response payloads in the Activity stream pane.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)

Get Weather Data

Use this recipe to get the weather data from an external source for a location and send the details to the caller as a response.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)

- [Activate and Run the Recipe](#)

About This Recipe

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 install the recipe and configure the connections within it. Subsequently, you can activate the integration in the recipe and send a REST request from an external application with the zip code of the location and the API key for the weather application to the integration's endpoint URL. When triggered, the integration flow fetches the weather data of the particular zip code and returns this information in the response.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- An account on an application such as OpenWeatherMap that provides weather data via REST APIs

Before You Install the Recipe

You must perform the following configuration tasks on OpenWeatherMap in order to connect to it using Oracle Integration, and retrieve weather information for a particular zip code with its weather data APIs.

To configure OpenWeatherMap:

1. Log in to OpenWeatherMap at <https://home.openweathermap.org>.
2. Click the user profile menu, and select **My API Keys**.
3. By default there is one API Key. You can generate a new key by clicking the **Generate** button.
4. Note down the value of the API Key.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the REST Weather Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the following URL: <code>https://api.openweathermap.org/data/2.5/weather</code>

3. In the Security section, select **No Security Policy**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a GET request to this endpoint URL along with the following request parameters:
 - **zip**
 - **appid**
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the values for the following request parameters:
 - **zip**: Enter the zip code of the location for which you want to get weather data
 - **appid**: Enter the value of the API key that you obtained in OpenWeatherMap. See [Before You Install the Recipe](#).
 - c. Click **Run**
 - d. In the Response section of the Configure and run page, you'll find a success status, 200 OK, and the response returned.

4. View the request and response payloads in the Activity stream pane.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Handle Throttling Using Parking Lot Pattern

Use this recipe to implement parking lot pattern in order to handle downstream throttling.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to use the parking lot pattern for handling downstream throttling and errors with an example use case of importing employee bank details from an external system to a downstream payroll system. An Oracle Autonomous Transaction Processing(ATP) database table is used for implementing the parking lot pattern.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. In this example, employee bank details are received from an external system such as Oracle HCM Cloud in a CSV file. The CSV file is then downloaded to a local directory on FTP Server or File Server. Subsequently, the employees bank details are registered in batches to an ATP database table and parked there for a specific time period before further processing. Thus, the integration flow has a chance to throttle the number of batches processed concurrently. The ready batches are picked up in a staged manner for processing and the processed data is finally updated to a downstream payroll system.

Overview of the recipe's integration flows:

- The **Request Persister** integration (**SR_BulkDownload_RequestPersister_ATP**) reads the CSV file received from external system and writes it to a local input directory in FTP. It reads the batch configurations (such as, group_id and group_type) and writes the metadata to the ATP database table. The registered batches are parked in the Parking Lot table in ATP for a specific period of time.
- The **Scheduled Dispatcher** integration (**SR_ScheduledDispatcher_CSVBatch**) is scheduled to run at required frequency. At every run, it fetches the batch requests from the Parking Lot table in ATP and dispatches them to the **Async Batch Processor** integration for processing. Thus, it triggers the **Async Batch Processor** integration to process the batches that are submitted for processing.
- The **Async Processor** integration (**SR_OneWay_Processor_HCM_To_Payroll**) processes all the submitted batch requests. Also, it updates the batch status (for example, if a batch is successfully processed, it updates the status to PROCESSED) in the Batch Statistics table in ATP. On successful processing, it invokes another integration (for example, **SR MOCK_EmpBankAccountProvider** integration in this recipe) to send the processed information to a downstream application via REST API calls. Any errors received in the response from the downstream application are written into an Error folder in FTP. These payload errors are also written to a Payload Error Record table in ATP.

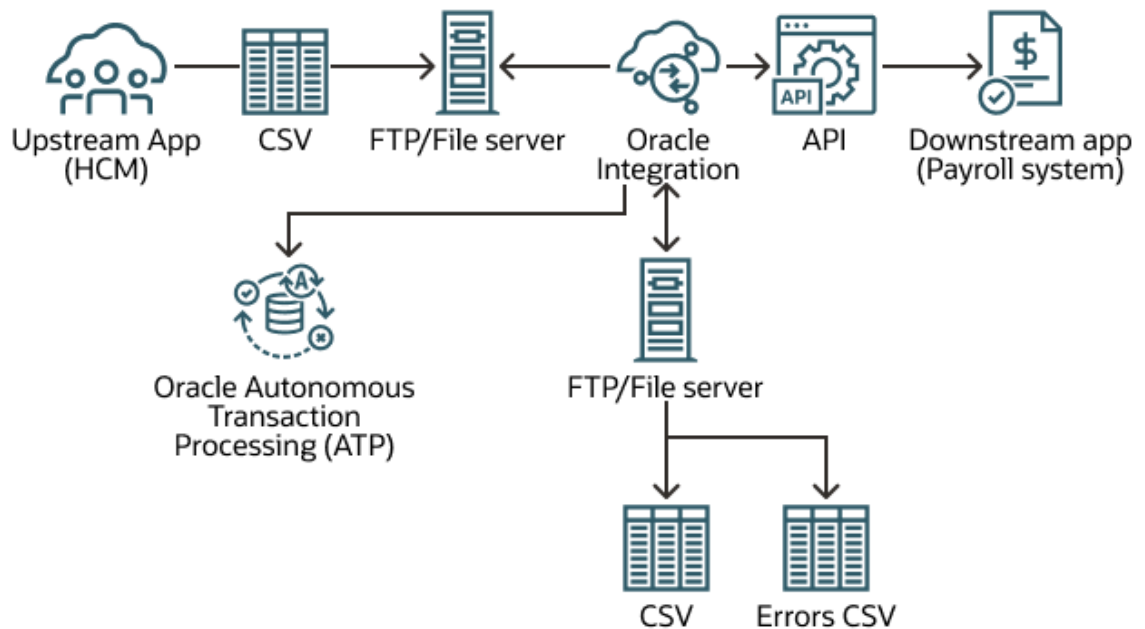
- The **Resubmission Processor** integration (**SR_ScheduledDispatcher_PayrollErrors**) is used to resubmit any batch requests failures/errors that were fixed. It fetches any record from the Payload Error Record table that are in READY state and re-submits them for processing.

System and Access Requirements

- Oracle Integration 3
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.



A CSV file with employee bank details from an upstream application such as Oracle HCM Cloud is downloaded to an FTP Server or File Server. The first integration flow (**Request Persister**) in Oracle Integration registers the batches in the ATP database. Each batch request is parked in the ATP database parking lot table for a certain period of time so the integration flow has a chance to throttle the number of batches processed concurrently. The second scheduled integration flow (**Scheduled Dispatcher**) in Oracle Integration fetches the batches from the parking lot table in ATP at a date and time of your choice, and dispatches the batches for further processing. The second integration triggers the third integration flow (**Async Processor**) which processes the dispatched batches. The processed batches are then sent to a downstream application such as a Payroll system via a REST API call. Any errors that are received as response from the downstream application are rectified and then resubmitted for processing by the fourth integration flow (**Resubmission Processor**) in Oracle Integration.

Before You Install the Recipe

You must perform the following configuration tasks on your FTP Server and Oracle Autonomous Transaction Processing instance in order to connect to these external systems using Oracle Integration and successfully handle throttling with the implementation of parking lot pattern.

Access Your FTP Server

Obtain a sFTP server and ensure that you're able to access it.

1. Log in to the FTP server using your user name and password through an FTP client; for example, FileZilla.
2. Create the following directories:
 - An input directory to place the CSV file received from the upstream system. For example: HCM_To_Payroll_Input. This file should be in a particular format for the recipe to run successfully. Download the following sample input file to successfully test the recipe: [Sample file](#).
 - A directory to place the files from the upstream system that were successfully processed. For example: HCM_Archive
 - A directory to receive files from the downstream system that could not be processed successfully. For example: Payroll_To_HCM_Input
 - A directory to archive files received from the downstream stream that were retried and processed successfully. For example: Payroll_Archive
 - A local input directory. For example: Input
 - A local output directory. For example: Output
 - A local error directory. For example: Error
 - A local processing directory. For example: Processing

You can also use File Server, an embedded sFTP server within Oracle Integration to store and work with the files. To use File Server, you must enable it for your Oracle Integration instance. See [Enable File Server](#) in *Using File Server in Oracle Integration 3*.

Configure Oracle Autonomous Transaction Processing

To configure Oracle Autonomous Transaction Processing, perform the following configuration tasks:

1. Complete the prerequisites for creating a connection from Oracle Integration to ATP. See [Prerequisites for Creating a Connection](#) in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.
2. Download and run the following SQL script to create the required parking lot table: [SQL Script](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources.

Configure the Oracle FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Name	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See [Configure Oracle Autonomous Transaction Processing](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC With OCI Signature .
Wallet	Select the check box, then click Upload to upload the wallet file. See Configure Oracle Autonomous Transaction Processing .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.
Object Storage Region	Specify the region in which your object storage is located. For example, us-ashburn-1.

Field	Information to Enter
Object Storage Tenancy OCID	Specify the value you copied from the Oracle Cloud Console. For example, <code>ocid1.tenancy.oc1.alphanumeric.value</code>
Object Storage Compartment OCID	Specify the value you copied from the Oracle Cloud Console. For example, <code>ocid1.compartment.oc1.alphanumeric.value</code>
Object Storage User OCID	Specify the value you copied from the Oracle Cloud Console. For example, <code>ocid1.user.oc1.alphanumeric.value</code>
Private Key	Click Upload to select the key you created. Ensure that the key is in RSA (PKCS1) format.

Note

Only a private key *without* a pass phrase/ password is supported.

Finger Print	Enter the finger print that was generated when you created the key in the Oracle Cloud Console.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Lookup Table

The recipe contains the following lookup tables that you can edit and configure as per your requirement.

- Email_Notification**
- Group_Setup**

- In the Lookups section, click the lookup name.
- Edit the following lookup tables.
 - In the **Email_Notification** lookup table, edit the value of the **BatchNotificationTo** lookup key by entering an email address of your choice to which notification error emails have to be sent.
 - In the **Group_Setup** lookup table, use the default directories provided for **GRPID: HCM_To_Payroll** or edit them as required with the directories that you created in your FTP server. See [Access Your FTP Server](#).
 - InputDir**: Enter the directory path to receive and store the input files from the upstream system (for example: `/pob-Workshop/HCM_To_Payroll_Input`).
 - ArchiveDir**: Enter the directory path to archive files from the upstream system which were successfully processed (for example: `/pob-Workshop/HCM_Archive`).
 - ProcessingHomeDir**: Enter the path of the home directory (for example: `/batchHome`).

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#)
2. Run the recipe.

Ensure that a sample input file with employee bank details in the correct format is uploaded to the input folder (in this case, HCM_To_Payroll_Input).

- a. Run the **SR_BulkDownload_RequestPersister_ATP** integration flow.
 - i. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - ii. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The integration now writes the input file to a local input folder (/Input). The information from the file, that is, the file metadata is persisted to the parking lot table (PAYLOAD_PARKING_LOT_TAB) in Oracle ATP.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

- b. Run the **SR_ScheduledDispatcher_CSVBatch** integration flow.
 - i. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - ii. On the Configure and run page, enter a value in the throttling parameter **MaxRecords_fromDB** to specify the number of records that should be fetched per run from the parking lot table (PAYLOAD_PARKING_LOT_TAB) in Oracle ATP.
 - iii. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The integration now reads the data from the parking lot table, and dispatches the data to the asynchronous integration flow **SR_OneWay_Processor_HCM_To_Payroll**, thus triggering the integration flow to process the batch records fetched from the parking lot table (PAYLOAD_PARKING_LOT_TAB). In the parking lot table, the status of the batch records that were submitted for processing changes from NEW to PROCESSED.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

The **SR_OneWay_Processor_HCM_To_Payroll** integration flow processes the batch files and calls the downstream application via a REST API call. It updates the batch

statistics (BATCH_STATISTICS) table in ATP with the status of each record that is being processed.

The records that the downstream application could not process are written to the payload error table (PAYLOAD_ERRORS_TAB) in ATP with the status as ERRORED. When the errors are rectified, the status of the records changes to READY.

The **SR_ScheduledDispatcher_PayrollErrors** integration flow can be run to fetch the READY records from the payload error table (PAYLOAD_ERRORS_TAB), process them, and send them to the downstream application.

3. Monitor the running of the integration flows in Oracle Integration. See [Monitor Integrations](#)

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Import a File into Oracle Content Management

This recipe imports a file into Oracle Content Management.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe receives a file as input and sends the file as an attachment to a specific folder in Oracle Content Management.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. The **Oracle REST Content Management Create Folder** integration receives folder name as input and creates a folder in Oracle Content Management. Subsequently, the **Oracle REST Content Management Upload File** integration receives file name, file content, and parent folder ID as input and imports the received file into the folder created earlier.

Note

You can only create one folder and import one file with each trigger.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Oracle Content Management
- An account on Oracle Content Management with the Administrator role

Before You Install the Recipe

To access Oracle Content Management using Oracle Integration and import a file, you must perform certain configuration tasks on your Oracle Content Management instance.

1. Create a parent folder in Oracle Content Management in which the new folder must be created. See **Create a new folder** in [Work with Your Files and Folders](#).
Note the `folderName`, `parentId`, and `description` of the parent folder.
2. Assign **Role for new public links** as **Contributor**. See **Change folder sharing settings** in [Work with Your Folders and Files](#).
3. Add a member to the folder. See **Add Folder Members** in [Share Folders](#).
4. Create an OAuth client and acquire the client ID and secret. See [Create an OAuth Client and Acquire a Client ID and Secret](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Trigger Connection

Use the following steps to configure the **ORCL-BRT-REST-ERP-CONNECTION** connection.

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

3. To return to the project workspace, click **Go back** .

Configure the Oracle REST Content Management Connection

Use the following steps to configure the **ORCL-BRT-CM-CONNECTION** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter your Oracle Integration instance URL. For instance, <code>https://<instance-name>.com/</code> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .

Field	Information to Enter
Access Token URI	Enter the access token URL of Oracle Content Management.
Client Id	Enter the client ID of the OAuth application created in Oracle Content Management. See Before You Install the Recipe .
Client Secret	Enter the client secret of the OAuth application created in Oracle Content Management.
Scope	Enter the scope of the OAuth application created in Oracle Content Management.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application to create a folder.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle REST Content Management Create Folder** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a GET request to the endpoint URL along with parameters such as `folderName`, `parentId`, and `description`. See [Before You Install the Recipe](#).

The recipe creates a folder in the parent folder in Oracle Content Management.

3. Run the recipe from an external application to import a file into the parent folder.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle REST Content Management Upload File** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the file details. Provide the text in the POST request's **Body** field.

Example request payload:

```
{
  "contenttype": "application/json",
  "filename": "<File name; For instance: Test.txt>",
  "FolderId": "<Parent ID from the earlier response>"
}
```

```
"FileContent": "<Enter file content in Base64 decode format>"  
}
```

The recipe creates the file with the file name and content you pass in the request body.

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to the Oracle Content Management and check for the new file and folder.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Import a Payment Request into Oracle ERP Cloud

Use this recipe to import a payment request from an FTP server to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports an interface file containing payment requests from an FTP server to Oracle ERP Cloud according to a schedule specified in Oracle Integration.

In Oracle ERP Cloud, a payment request is used to make a one-time payment (without creating a supplier record) to external suppliers or parties. When a payment request (transaction) is imported into Oracle ERP Cloud, the payees, parties, and bank accounts are automatically created in the application, thereby simplifying the task of making the payment. In addition, the transactions are validated automatically, eliminating the need for manual intervention.

As part of this recipe, Oracle provides an example file structure that you can use to load a payment request into Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the recipe manually or specify an execution schedule for it. When triggered, the recipe loads a payment request from a specified FTP server to Oracle ERP Cloud.

System and Access Requirements

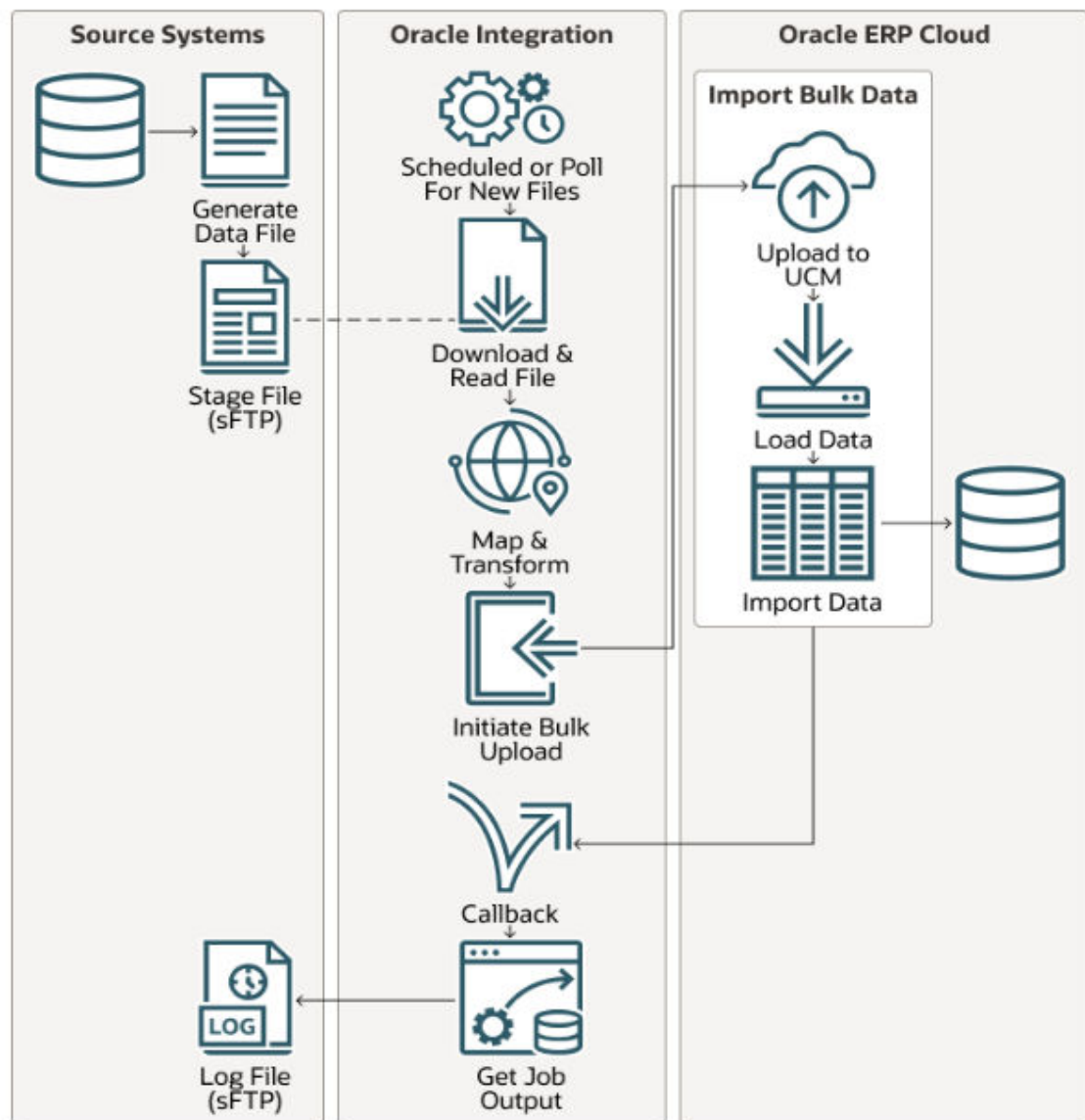
- Oracle Integration, Version 21.4.2.0.0 or higher
- Oracle ERP Cloud, Version 21d or higher
- Accounts on Oracle ERP Cloud with the Administrator role and the Accounts Payable Invoice Supervisor/Accounts Payable Manager/Accounts Payable Specialist role (in order to check for payment requests)
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server.

Recipe Schema

This section provides an architectural overview of the recipe.

The primary integration flow (**Oracle FTP ERP Cloud Payment Request**), when triggered by an execution schedule or a manual submission, queries the FTP server for a payment request file (.zip). If a file is found, the integration flow processes the file. Initially, the flow maps the file's layout to the layout required by Oracle ERP Cloud, and then it imports the payment requests in bulk through the ERP Adapter.

After importing the payment request, Oracle ERP Cloud makes a callback to the secondary integration flow (**Oracle FTP ERP Cloud Payment Request Callback**), which downloads the log files from Oracle ERP Cloud and places them on the FTP server.



Payment Request File Definition

This section describes the file layout used by the primary integration flow (**Oracle FTP ERP Cloud Payment Request**) of the recipe. This integration flow is configured to look for a .zip file on your sFTP server. The .zip file must contain the following two files:

- apPaymentRequest.properties
- apPaymentRequest.csv

Definition: apPaymentRequest.properties

This file tells Oracle ERP Cloud how to process and validate the payment requests. This file is in the .csv format.

Parameter	Value
Program Location	oracle/ apps/ess/financials/payables/invoices/ transactions/
Program	APXPRIMPT
Interface	The file name to be loaded, this will be the the name placed within the FTPImportFileName lookup without the .zip suffix. For example: apPaymentRequest.
Business Unit	Setup and Maintenance/Manage Business Unit.
Accounting Date	Enter an accounting date. If you don't enter a date, the Accounting Date is based on the Accounting Date Basis option.
Oracle_Int1	Hard-coded internal value set to 1000.
Source	For Payment Requests, this is always ONE_TIME_PAYMENTS.
Import Set	Name of an import set to limit the import.
Purge	Yes, for Payables to delete records that match the Source and Group ID of this import. No, to keep all the records except the rejected records.
Summarise	Enter whether to summarize the audit portion of the Payables Payment Request Interface Import Report.
Ledger	Setup and Maintenance/Manage Primary Ledgers.
Invoice Group	Enter the name of an invoice group.
Oracle_Int2	Hard-coded internal value set to 1.

Example Values: apPaymentRequest.properties

The following table provides an example set of values for the parameter file. Ensure that you enter the values specific to your configuration in the Interface, Business Unit, and Ledger fields.

Parameter	Value
Program Location	oracle/apps/ess/financials/payables/invoices/ transactions/
Program	APXPRIMPT
Interface	apPaymentRequest
Business Unit	300000047498175

Parameter	Value
Accounting Date	null
Oracle_Int1	1000
Source	ONE_TIME_PAYMENTS
Import Set	null
Purge	N
Summarise	N
Ledger	300000047488112
Invoice Group	null
Oracle_Int2	1

Example File: apPaymentRequest.properties

This section provides an example parameter file, with the name: `apPaymentRequest.properties`.

The file name must be same as the value entered in the Interface field within the file, with `.properties` as suffix.

Example File:

```
ProgramLocation,Program,Interface,BusinessUnit,AccountingDate,Oracle_Int1,Source,
ImportSet,Purge,Summarise,Ledger,InvoiceGroup,Oracle_Int2 /oracle/apps/ess/
financials/payables/invoices/
transactions/,APXPRIMPT,apPaymentRequest,300000047498175,null,1000,ONE_TIME_PAYME
NTS,null,N,N,300000047488112,null,1
```

Definition: apPaymentRequest.csv

This file contains the payment requests that are paid through Oracle ERP Cloud. The file provided here is an example, containing the minimum set of data required to load a payment request.

Prepare this file in the `.csv` format, using the following structure:

Parameter	Description
Transaction Identifier	Identifier that uniquely identifies a payment request line record.
Party Name	Name of the party to whom the payment is to be made.
Party Original System Reference	Party reference from the original system or external source.
Party Type	Type of party receiving the payment.
Location Original System Reference	Original system reference for the party site address.
Country	Location of the address by country.
Address Line 1	First line of the party address.
Address Line 2	Second line of the party address.
City	City of party to whom the payment is to be made.
State	State of party to whom the payment is to be made.
Province	Province of party to whom the payment is to be made.
County	County of party to whom the payment is to be made.
Postal Code	Postal code of party to whom the payment is to be made.

Parameter	Description
Postal Plus 4 code	Four-digit extension to postal code.
Addressee	Person or recipient located at the address.
Global Location Number	Number that uniquely identifies each global location in a trading-partner enterprise.
Language	Operating language of the party-site address.
Phone Country Code	Country code for the telephone number.
Phone Area Code	Area code for address phone number.
Phone	Phone number of the party receiving payment.
Phone Extension	Phone extension of the party receiving payment.
Payee E-Mail address	E-mail address of the payee to whom the payment is to be made.
Party Tax Registration Number	Unique identifier assigned to a party or party site by a tax authority.
Account Country Code	Country code of the bank account of payee.
Account Currency Code	Currency code of the bank account of payee.
Account Number	Bank account number of the payee.
IBAN	International Bank Account Number of payee.
Secondary Account Reference	Account number reference such as Building Society Roll Number.
Account Suffix	One to three digit number at the end of the account number that determines the kind of account.
Account Name	Name of the payee bank account.
Account Type Code	Type of bank account, such as SAVINGS and CHECKING.
Bank Name	Name of the payee bank to whom the payment is to be made.
Bank Number	Number of the payee bank to whom the payment is to be made.
Bank Branch Name	Name of the branch of the payee bank.
Bank Branch Number	Number of the bank branch. This is known as the routing transit number in US and Canada.
BIC	SWIFT or Bank Identification Code (BIC) of the bank branch.
Check Digits	One or more digits used to validate a bank account number.
Account Description	Description of the payee bank account.
Account Alternate Name	Alternate account name of the bank account of payee.
Business Unit	Sold-to business unit name for which the transaction is being created.
Source	Name of the external system from which a payment request is imported.
Invoice Number	Unique number for payment request.
Invoice Date	Date on which payment request is submitted.
Currency	Currency of the payment request.
Description	Information describing the invoice created for payment.
Import Set	Name or number assigned to the import set containing payment requests.
Legal Entity	The name of the legal entity that belongs to your own corporate structure (enterprise).
Payment Terms	Name of the payment terms used to create installments and to calculate due dates, discount dates, and discount amounts for each invoice.

Parameter	Description
Payment Method	Code identifying the payment method.
Pay Group	Used to group payment requests into a single pay run.
Liability Combination	Account to which liability is recorded in accrual basis of accounting.
Line Number	The number of the invoice line.
Amount	The invoice line amount to import for payment.
Line Description	Line description.
Distribution Combination	Account combination used for creating invoice distribution.

Example Values: apPaymentRequest.csv

The following table provides an example set of values for the payment-request file.

Parameter	Description
Transaction Identifier	100109
Party Name	Fred Jones
Party Original System Reference	OTP-PTY-100109
Party Type	ORGANIZATION
Location Original System Reference	OTP-PTY-100109
Country	GB
Address Line 1	Oracle Parkway
Address Line 2	Thames Valley Park
City	Reading
State	UK
Province	N/A
County	N/A
Postal Code	RG6
Postal Plus 4 code	1RA
Addressee	N/A
Global Location Number	N/A
Language	N/A
Phone Country Code	N/A
Phone Area Code	N/A
Phone	N/A
Phone Extension	N/A
Payee E-Mail address	fred.jones@emaildomain.com
Party Tax Registration Number	N/A
Account Country Code	GB
Account Currency Code	GBP
Account Number	99453336
IBAN	N/A
Secondary Account Reference	N/A
Account Suffix	N/A
Account Name	F Jones

Parameter	Description
Account Type Code	N/A
Bank Name	HSBC
Bank Number	N/A
Bank Branch Name	Reading
Bank Branch Number	754322
BIC	N/A
Check Digits	N/A
Account Description	N/A
Account Alternate Name	N/A
Business Unit	UK Business Unit
Source	ONE_TIME_PAYMENTS
Invoice Number	OTP-FJONES-109
Invoice Date	2021/11/09
Currency	GBP
Description	Refund
Import Set	GBP, Refund, , UK Legal
Legal Entity	UK Legal Entity
Payment Terms	Immediate
Payment Method	N/A
Pay Group	N/A
Liability Combination	303.30.2210.000.000.000
Line Number	1
Amount	20
Line Description	One time payment lines
Distribution Combination	303.30.7740.000.000.000

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration and load payment requests.

Access Your FTP Server and Upload the Files

Obtain an SFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server from which to read the files. Note the path of this directory.
3. Upload a payment-request file (.zip) into the directory you've created. You must name the file as `apPaymentRequest.zip`.

Note

The .zip file must contain two files by the name, apPaymentRequest.properties and apPaymentRequest.csv. For details on the structure of these files, see [Payment Request File Definition](#).

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see [Assign Required Roles to an Integration User](#).
 - Integration Specialist
 - AttachmentsUser
 - SOAOperator
 - FND_MANAGE_CATALOG_SERVICE_PRIV
 - Payables Invoice Creation

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your SFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Public Key Authentication..
User Name	Enter the user name to connect to your SFTP server.

Field	Information to Enter
Private Key File	Upload the private key file for your SFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table to specify the FTP server directory from which the file must be read, the name of the file to be read, and an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. In the lookup table, enter the following information.

Property Name	Information to Enter
FTPImportFileName	Specify the name of the file to be read from the FTP server. For example, <code>apPaymentRequest</code> . See Access Your FTP Server and Upload the Files .
emailErrorNotificationTo	Enter an email ID of your choice.
FTPImportDirectory	Specify the path of the directory on your FTP server from where the file must be read. For example: <code>/home/user2/FTP</code> .

3. Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow (**Oracle FTP ERP Cloud Payment Request**), then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check for the new payment request on Oracle ERP Cloud.
 - a. Log in to the Oracle ERP Cloud instance using a separate account, for example, as a user with the Accounts Payable Invoice Supervisor/Accounts Payable Manager/Accounts Payable Specialist role.
 - b. Navigate to the Invoice Dashboard screen.
 - c. In the **Invoice Number** search field, enter the value of the **Invoice Number** field that's in your interface file, and click **Search**. The imported payment-instruction invoice is displayed.
5. On your FTP server, check the log files from Oracle ERP Cloud.
 - a. Log in to the server and navigate to your directory.

You'll find a file with the following name: `apPaymentsRequest.zip_dateandtimr.zip`.
 - b. Download this file from the FTP server to your local system.
 - c. Unzip the file and inspect the log files present within.

Related Documentation

- [Using the FTP Adapter with Oracle Integration 3](#)
- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)
- [File-Based Data Import \(FBDI\) for Financials](#)
- [How Payables Payment Request Import Data is Processed](#)
- [Definition of Available Fields in the Payment Request Integration](#)

Import Absence Records from Oracle HCM Cloud to Google Calendar

Use this recipe to import approved absence records from Oracle HCM Cloud to Google Calendar.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports approved absence records from Oracle HCM Cloud to Google Calendar as per a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

When triggered, the integration flow reads all the approved absence records in Oracle HCM Cloud and creates subsequent events in Google Calendar. An email notification is sent to the individual event participants.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Google Calendar
- An account on Google Calendar with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Google Calendar instances in order to successfully connect to these external systems using Oracle Integration and import approved absence records.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and read the approved absence records, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Also, create and approve absence records.

1. Create absence records in the Oracle HCM Cloud instance. See [Add Absences](#).

2. Approve the absence records. See [Absence Administration Tasks](#).

Configure Google Calendar

To connect to Google Calendar from Oracle Integration and create events, you must create an OAuth application on your Google Calendar and obtain access keys for the application. See *Prerequisites for Creating a Connection in Using the Google Calendar Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Google Calendar Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the Client ID obtained when you configured Microsoft Office 365 Calendar. See Configure Google Calendar .
Client Secret	Enter the Client Secret obtained when you configured Google Calendar.

Field	Information to Enter
Scope	Enter https://www.googleapis.com/auth/calendar

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.
You'll now be redirected to the Google Calendar page.
6. Enter your Google Calendar account credentials.
A page is displayed asking for permission to interact with the account.
7. Click **Yes**.
You're informed that access is allowed. You can now switch back to the Google Calendar Connection window of Oracle Integration to test your connection.
8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
9. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **NotificationLookup** lookup table to configure the email address to which the notification emails are sent.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

Key	Value
FromAddress	Specify the email address from which the notification emails are sent. Enter an email address of your choice.
ToAddress	Specify the email address to which the notification emails are sent. Enter an email address of your choice.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **Oracle HCM Google Create Absence Calendar Event** integration flow. You can update the integration properties only after activating this flow.

- a. In the Integrations section, click **Actions** . . . on the integration flow, then select **Update property values**.
- b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
CalendarTimeZone	Specify the time zone in which the calendar event must be created.
Attendees	Specify the email address of the event attendees. Enter the email address of your choice.
CalendarTimeAdd	Specify the UTC time difference. For example, +07:00.

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

3. Run the recipe.

- a. In the Integrations section of the project workspace, click **Actions** . . . on the integration flow, then select **Run**.
- b. On the Configure and run page, you can specify a value for the `lastRunDate` parameter, and click **Run**.

The `lastRunDate` parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Oracle HCM Cloud approved absence records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, MM-DD-YYYY. To fetch all the approved absence records from the beginning, set the value as zero.

- c. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Google Calendar instance and check for new events created for the approved absence records. See [Search on Google Calendar](#).

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the Google Calendar Adapter with Oracle Integration 3*

Import Absence Records from Oracle HCM Cloud to Microsoft Calendar

Use this recipe to import approved absence records from Oracle HCM Cloud to Microsoft Office 365 Calendar.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports approved absence records from Oracle HCM Cloud to Microsoft Office 365 Calendar as per a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

When triggered, the integration flow reads all the approved absence records in Oracle HCM Cloud and creates subsequent events in Microsoft Office 365 Calendar. An email notification is sent to the individual event participants.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Microsoft Azure
- An account on Microsoft Azure with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Microsoft Office 365 Calendar instances in order to successfully connect to these external systems using Oracle Integration and import approved absence records.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and read the approved absence records, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Also, create and approve absence records.

1. Create absence records in the Oracle HCM Cloud instance. See [Add Absences](#).

2. Approve the absence records. See [Absence Administration Tasks](#).

Configure Microsoft Office 365 Calendar

To connect to Microsoft Office 365 Calendar from Oracle Integration and create events, you must create an OAuth application on your Microsoft Office 365 Calendar and obtain access keys for the application. See *Prerequisites for Creating a Connection in Using the Microsoft Office 365 Calendar Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

Use the following steps to configure the **Oracle HCM Cloud Connection** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Microsoft Office 365 Calendar Connection

Use the following steps to configure the **Oracle Microsoft Office 365 Calendar Connection** connection.

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the Client ID obtained when you configured Microsoft Office 365 Calendar. See Configure Microsoft Office 365 Calendar .
Client Secret	Enter the Client Secret obtained when you configured Microsoft Office 365 Calendar.
Scope	Enter https://graph.microsoft.com/ Calendars.ReadWrite offline_access

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.
You'll now be redirected to the Microsoft Office 365 Calendar page.
6. Enter your Microsoft Office 365 Calendar account credentials.
A page is displayed asking for permission to interact with the account.
7. Click **Yes**.
You're informed that access is allowed. You can now switch back to the Microsoft Office 365 Calendar Connection window of Oracle Integration to test your connection.
8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
9. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **Oracle HCM Microsoft Create Absence Calendar Event** integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
CalendarTimeZone	Specify the time zone in which the calendar event must be created.
Attendees	Specify the email address of the event attendees. Enter the email address of your choice.

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, you can specify a value for the `lastRunDateTime` parameter, and click **Run**.

The `lastRunDateTime` parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Oracle HCM Cloud approved absence records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the approved absence records from the beginning, set the value as zero.

- c. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to Microsoft Office 365 Calendar and check for new events created for the approved absence records.
 - a. Sign in to [Outlook.com](#).
 - b. Select the **Calendar** icon.
 - c. In the Search box, type the keywords you want to search and check for new events. See [Search for calendar items](#).

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the Microsoft Office 365 Calendar Adapter with Oracle Integration 3*

Import and Update Files in Bulk

Use this recipe to imports a set of files, updates their content, and writes them back to an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe 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.

To use the recipe, you must install the recipe and configure the connection and other resources within it. For the recipe to successfully read and transform the files, you must provide the schema files (.CSV or .txt files) that contain the required data structures in the Stage File actions for reading and writing files.

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 3*](#).

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- A secure FTP (sFTP) server or File Server
- An FTP client to access the external sFTP server

Before You Install the Recipe

Perform the following tasks on your sFTP server.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server from which to read input files and write the transformed output files back.
3. Place a sample zip file along with the comma separated value (CSV) files in the directory.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

- In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
- In the Update property values panel, update the integration properties with appropriate values.
 - directory**: This integration property holds the path of the directory that you created in your FTP server. See [Before You Install the Recipe](#).
 - inputfilename**: This integration property holds the input file name.
 - outputfilename**: This integration property holds the output file name.
- Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the recipe.
 - In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

- Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
- Log in to the FTP server and check if the output file has been uploaded to the specified directory.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using File Server in Oracle Integration 3*

Import Candidate Records from an FTP Server to Taleo

Use this recipe to import candidate records from an FTP server to Taleo.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports candidate records from an FTP server to Taleo as per a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

The recipe contains the following integrations:

- **Oracle FTP Taleo Import Candidate:** The scheduled parent integration that lists all the input files and triggers the child integration with a specific file name.
- **Oracle Taleo Import Candidate:** The app-driven child integration that processes the file content and imports candidate records into Taleo. Further, the recipe adds the result to the Result FTP folder and moves the input file to the Archive folder.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server
- Taleo
- An account on Taleo with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and Taleo instance in order to successfully connect to these external systems using Oracle Integration and import candidate records.

Access Your FTP Server and Upload the Files

Obtain an sFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.

2. Create an individual directory on the server to store the following files. Note the path of the directories.
 - a. Input files (Input folder)
 - b. Result files (Result folder)
 - c. Archive files (Archive folder)
3. Upload a Candidates Record CSV file to the Input folder in the following format:

```
EmployeeNumber,SSN,firstname,miidialinitial,lastname,email,address,address2,cit  
y,zip,Salary,NotifyByEmail,phone,Hourly498062,645-81-5006,FirstName,MiddleInit  
ial,LastName,oh_test2@le.com,258 Mudisa  
Point,Address2,Urtaki,18996,725795,false,(415)228-5682,false
```

Configure Taleo

To access the Taleo instance from Oracle Integration and import candidate records, you'll need to create a user account for Oracle Integration on Taleo and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Import Candidates Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Taleo Import Candidates Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Instance URL	Enter the instance URL of the Taleo instance. For example, <code>https://<instance-name>.taleo.net/</code> .
Port	Enter the port of the Taleo instance. For example, 443.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Taleo. See Configure Taleo .
Password	Enter the password of the account created for Oracle Integration on Taleo.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Import Candidates Connection

1. In the Connections section, click the connection name.
2. Click **Save**. If prompted, click **Save** again.
3. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **Oracle Taleo Import Candidate** integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

- **FromAddress:** Specify the email address from which the notification emails are sent. Enter an email address of your choice.
 - **ToAddress:** Specify the email address to which the notification emails are sent. Enter an email address of your choice.
- c. Click **Submit**.
- You get a confirmation message that the integration properties have been updated successfully.
3. Update property values for the **Oracle FTP Taleo Import Candidate** integration flow. You can update the integration properties only after activating this flow.
- a. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
ResultFolder	Specify the path of the Results FTP folder in which the result files must be placed. See Access Your FTP Server and Upload the Files .
FromAddress	Specify the email address from which the notification emails are sent. Enter an email address of your choice.
ArchiveFolder	Specify the path of the Archive FTP folder in which the input files must be archived.
InputFolder	Specify the path of the FTP folder in which the input files must be listed.
ToAddress	Specify the email address to which the notification emails are sent. Enter an email address of your choice.

- c. Click **Submit**.
- You get a confirmation message that the integration properties have been updated successfully.
4. Run the recipe.
- a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
- You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Log in to your Taleo instance and check for new candidate records. See [Candidate Management](#).

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration 3*

Import Contacts from Oracle Autonomous Transaction Processing to Oracle Responsys

Use this recipe to import contact details from Oracle Autonomous Transaction Processing (ATP) database to Oracle Responsys.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application driven recipe and is triggered whenever customer details are created in ATP database using SQL Developer. Subsequently, a corresponding customer details record is created in Oracle Responsys.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a customer's contact information is created in ATP database using SQL Developer, the integration flow is triggered, and it creates a corresponding customer contact record in Oracle Responsys. Basic information such as customer ID, email, first name, and last name are synchronized between the Oracle Autonomous Transaction Processing and Oracle Responsys platforms.

System and Access Requirements

- Oracle Integration, Version 24.08 or higher
- Oracle Responsys
- An account on Oracle Responsys with the Administrator role
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role
- Oracle SQL Developer

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle SQL Developer instance to successfully connect to Oracle Integration and import customer contact details from Oracle Autonomous Transaction Processing to Oracle Responsys.

Log in to Oracle SQL Developer instance as an **Administrator** and perform the following tasks.

1. Create a new connection to ATP database using Oracle SQL Developer
 - a. Under the **Connections** pane on the left, click the plus sign **+** to create a new connection.
 - b. In the New/Select Database Connection window, enter the following details:

- i. Enter the **Name** of the connection.
- ii. Enter the **Username** and **Password**.
- iii. Select **Cloud Wallet** from the **Connection Type** drop-down list.
- iv. Click **Browse** next to the **Configuration File** field and upload the Oracle Autonomous Transaction Processing wallet file. To obtain the wallet file, see Prerequisites for Creating a Connection in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.
- v. Enter the **Service**.
- vi. Click **Test** to check if the connection is successful.
- vii. Click **Save**.

The new connection appears under the Connection pane.

2. Create a new database table.
 - a. Under the Connection pane, expand the new connection to load all the objects under it.
 - b. Right-click **Tables (Filtered)** and select **New Table** from the menu that displays.
 - c. In the Create Table window, enter the name of the table in the **Name** field.
 - d. Define the Name and Datatype columns in the table by entering the following information.

Column Name	Datatype
CUSTOMERID	VARCHAR2(100 BYTE)
EMAIL	VARCHAR2(100 BYTE)
FIRSTNAME	VARCHAR2(100 BYTE)
LASTNAME	VARCHAR2(100 BYTE)

- e. Commit the changes.
3. Send the data from the newly created database table to a queue table by running the following script: [Oracle_SQL_Developer_script](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the AQ Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host	Not required if you are using Oracle Wallet as the security policy.

Field	Information to Enter
Port	Not required if you are using Oracle Wallet as the security policy.
SID	Not required if you are using Oracle Wallet as the security policy.
Service Name	Enter the logical representation of a service used for client connections.

Note

The Service Name should be kept low profile and should be same as the `tnsnames.ora` file in the wallet. To know more, see Service Name Specified on the Connections Page is Too Long in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Oracle Wallet .
Wallet	Select the checkbox and click Upload to upload the wallet file in .zip format. See the Prerequisites to Use the Oracle Wallet Security Policy section in Prerequisites for Creating a Connection in <i>Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration 3</i> .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database username to connect to the Oracle Advanced Queuing (AQ) server account.

Note

The Database Service Username is not the same as the Service Name which you specified in the Properties section above.

Database Service Password	Enter the database password to connect to the Oracle Advanced Queuing (AQ) server account.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle Responsys Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the Oracle Responsys log in URL.

Note

The URL is provided by Oracle Support Services when Web Services is enabled on your account.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Custom Security Policy .
Username	Enter the username provided by Oracle Support Services once Web Service is enabled on your account.
Password	Enter the Password provided by Oracle Support Services once Web Service is enabled on your account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORCL-BRT-RESPONSYS_PROPERTIES** lookup table to map ATP customer details to Oracle Responsys customer details.

1. In the Lookups section, click the lookup name.
2. The **ORCL-BRT-RESPONSYS_PROPERTIES** lookup table is used to store customer contact details. Edit it as necessary.

Key	Description	Example
HtmlValue	Value of incoming preferred email format data. For example, H may represent a preference for HTML formatted email.	H
OptinValue	Value of incoming opt-in status data that represents an opt-in status. For example, 1 may represent an opt-in status.	1
TextValue	Value of incoming preferred email format data. For example, T may represent a preference for Text formatted email.	T
InsertOnNoMatch	Indicates what should be done for records where a match is not found (true = insert / false = no insert).	true
UpdateOnMatch	Controls how the existing record should be updated.	REPLACE_ALL
MatchColumnName1	First match column for determining whether an insert or update should occur.	EMAIL_ADDRESS_

Key	Description	Example
MatchOperator	Controls how the boolean expression involving the match columns is constructed to determine a match between the incoming records and existing records.	NONE
DefaultPermissionStatus	Controls how the boolean expression involving the match columns is constructed to determine a match between the incoming records and existing records.	OPTIN

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Update Integration Properties

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **EmailTo**: This integration property holds the email address to which error notifications are sent.
 - b. **retryCount**: This integration property holds the number of retries when an invoke fails. The default value is 3.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

Note

Ensure that you create customer contact table in ATP database and queue its data into a queue table before activating and running the recipe. See [Before You Install the Recipe](#).

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle Responsys instance and check for the new contact information.
 - a. Navigate to **Data** and then to **Profile List**.
 - b. Select **List_EF** in the **Change List** drop-down list.
 - c. In the List Information window that appears, click **View Records**.

An AS/List_EF window loaded with imported data appears where you can view the successfully imported customer contact details from ATP.

Related Documentation

- *Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration 3*
- *Using the Oracle Responsys Adapter with Oracle Integration 3*

Import Customer Records from an FTP Server to Oracle ERP Cloud

Use this recipe to import customer records in bulk from an FTP server to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports customers from an FTP server location to Oracle ERP Cloud, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the zip file containing Comma Separated Values (CSV) from a specified location in FTP and imports them as customer records to your Oracle ERP Cloud instance.

You can also store the zip file to import and work with on File Server, an embedded sFTP server within Oracle Integration. To use File Server, you must enable it for your Oracle Integration instance. See [Enable File Server in Using File Server in Oracle Integration 3](#).

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle ERP Cloud

- An account on Oracle ERP Cloud with the Administrator role
- An FTP server
- An FTP client to access the FTP server

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance and FTP server in order to successfully connect to these external systems using Oracle Integration and import customer records.

Access Your FTP Server

Obtain an FTP server and ensure that you can access it.

1. Sign in to the server using your username and password through an FTP client; for example, FileZilla
2. Create a directory on the server to store the CSV zip file. Note down the path of the directory.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Also, generate the CSV zip file with customer records, and update ESS job definition fields.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user
3. Generate the CSV zip file.
 - a. Navigate to **Receivables**, select **Billing**, and then select **Customers**.
 - b. Click **Upload Customers from Spreadsheet**.
 - c. In the Manage Customer Uploads page, click **Download Customer Spreadsheet Template**.
 - d. Populate the template with all the customer details. Further, follow the instructions in the template to generate and download the CSV zip file.
4. Update the ESS job definition fields.
 - a. Navigate to **My Enterprise** and select **Setup and Maintenance**.
 - b. Choose **Financials** from the **Subject Area** drop-down menu. Further, click **Search**.

- c. In the search box, search for **Manage Enterprise Scheduler Job Definitions and Job Sets for Financial, Supply Chain Management, and Related Applications**, and select the same from the search results.
- d. On the resulting page, navigate to the **Manage Job Definitions** tab.
- e. Update the following ESS job definition fields.

Field	Description	Information to Enter
JobDefinitionName	The name of the ESS job	UploadCustomerEss
JobPackageName	The name of the package containing the ESS Job	/oracle/apps/ess/financials/receivables/customerSetup/customerProfileClasses/

- f. Update the parameters for each ESS job definition fields. Navigate to the **UploadCustomerESS: Parameters** tab to view and edit the parameters.

Parameter	Description
Batch Name	Enter the name of the batch in which the customers must be created in bulk
Data File	Enter the name of the file being imported
Batch ID	Enter the unique ID of the batch.
ATTRIBUTE4	Specify the action. For example, CREATE.

Note

If you prefer to customize any of the above-mentioned values, ensure that you make necessary changes while updating the integration properties.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. 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. In the navigation pane, click **Settings**, then **File Server**, then **Settings**.
 - b. From the **General** section, copy the IP and port.
2. In the Connections section, click the connection name.
3. In the Properties section, enter the following details:
 - a. **FTP Server Host Address**: the IP for your external FTP file server or the IP you copied from the File Server settings. For instance, server123.company_name.com.
 - b. **FTP Server Port**: the port for your external FTP file server or the port you copied from the File Server settings. For instance, 21.

- c. **SFTP Connection:** specify the value as No.
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 FTP server.
Password	Enter the password to connect to your FTP server.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
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.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
ESSJobDefinitionName_Param	Specify the name of the ESS job definition.
ESSJobPackageName_Param	Specify the name of the package that contains the ESS job definition.
ESSJOB_Wait_Count_Param	Specify the number of minutes the integration must wait for the ESS job to be completed.
ftpOutputDir	Specify the path of the FTP directory where you can download the job run log files.
ftpInputDir	Specify the path of the FTP directory where the integration must import the CSV zip file with customer records. See Access Your FTP Server .

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

3. Log in to your FTP server and upload the CSV zip file with the customer data in the corresponding directory.
4. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Log in to Oracle ERP Cloud and check for new customer records.
 - a. Navigate to **Receivables**, select **Billing**, and then select **Manage Data Import**.
 - b. Click the corresponding **Batch ID** to view details such as total number of records, records processed, and error records.

You can also view individual customer records. To do so, navigate to **Billing Work Area**, select **Manage Customers**, and search for the specific customer record.

Related Documentation

- [Using the FTP Adapter with Oracle Integration 3](#)
- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)

Import Customer Records from Shopify to Stripe

Use this recipe to import customer records from Shopify to Stripe.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)

- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a corresponding customer record in Stripe whenever a customer record is created in Shopify.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a customer record is created in the Shopify instance, the integration flow of the recipe is triggered, and it creates a corresponding record in Stripe. Basic data, such as customer name, city, state, postal code, country, email, and phone, are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Shopify
- An account on Shopify with the Administrator role
- Stripe
- An account on Stripe with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Shopify and Stripe instances in order to successfully connect to these external systems using Oracle Integration and transfer customer records between them.

Configure Shopify

To configure Shopify, see Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Obtain Stripe API Keys

To access the Stripe instance from Oracle Integration, you'll require the API key associated with your Stripe account.

To obtain your Stripe account's API key:

1. Sign in to your Stripe instance as an administrator.
2. In the left navigation pane, click **Developers**, then **API keys**.

The API keys page is displayed with all the keys associated with your account.

3. In the Standard keys section, click **Reveal test key** in the **Secret key** row.
4. Note the secret key.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Shopify Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the Shopify host name. The Shopify host name appears in the Shopify home URL. For example: <code>https://hostname.myshopify.com</code> .
Shopify REST API Version	Enter the REST API version.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Shopify Security Policy selected.
Username	Enter the API key obtained earlier. See Configure Shopify .
Password	Enter the password obtained earlier.
Shared secret	Enter the shared secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Stripe REST Trigger Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the following URL: <code>https://api.stripe.com/</code> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave API Key Based Authentication selected.
API Key	Enter the API key associated with your Stripe account. See Obtain Stripe API Keys .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must create a customer record in Shopify.

- a. Log in to the Shopify instance.
- b. On the left navigation pane, click **Customers**.
- c. In the Customers window, click **Add Customer**.
- d. Enter the following details, and click **Save**.
 - First Name, Last Name, Email, Phone, and Address.
 - Select the **Collect Tax** check box if the customer is not tax exempted.
 - Add additional notes if necessary.
- e. Click **Create Order**.

You've now triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Stripe instance and check for the new customer record created. On the Stripe instance, click the **Customers** tab. The list of customers is displayed, which contains the corresponding customer record for the record created in Shopify.

Related Documentation

- [Using the Shopify Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)

Import Customer Records in Bulk from Oracle JD Edwards to Salesforce

Use this recipe to import customer records from Oracle JD Edwards as accounts in Salesforce.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports customer records from Oracle JD Edwards as accounts in Salesforce, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the customer records in Oracle JD Edwards and imports them as accounts to your Salesforce instance.

Note

Ensure that your Oracle JD Edwards instance has customer records that are to be imported as accounts to your Salesforce instance.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- Oracle JD Edwards
- An account on Oracle JD Edwards with the Administrator role
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle JD Edwards and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and import customer records in bulk.

Configure Oracle JD Edwards

To access the Oracle JD Edwards instance from Oracle Integration, you must create an Oracle JD Edwards agent group and download and run the connectivity agent installer to install the agent in your local environment. See *Download and Run the Connectivity Agent Installer in Using Integrations in Oracle Integration 3*.

Configure Salesforce

To access Salesforce from Oracle Integration, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create a custom field for Salesforce lead records to store Oracle JD Edwards Customer ID, Oracle JD Edwards User ID, and so on.

Log in to your Salesforce instance as an **Administrator** and perform the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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. In the **Existing Profile** field, select **Standard User**.
 - ii. Enter a name for the new profile, for example, **API Enabled**, and click **Save**.

The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user. For example, **Integration User05**.
 - ii. In the **Email** field, enter a valid email address.

The email address you enter is automatically populated in the **Username** field. Note this username.
 - iii. In the **User License** field, select **Salesforce**.

- iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
- v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.
Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the **Quick Links** section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account. Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
- 3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
- 4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
- 5. Create custom fields for Salesforce account records to hold the Oracle JD Edwards IDs. This recipe uses unique IDs associated with customer records in Oracle JD Edwards to import Oracle JD Edwards customers as Salesforce accounts.
 - a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Customize**, then **Accounts**, and then click **Fields**.
 - b. On the Accounts Fields page, scroll to the **Account Custom Fields & Relationships** section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:

- Enter `JDEdwardsCustomerId` 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.
 - 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 Account Fields page, you can see the new field added under the **Quote Custom Fields & Relationships** section.

Similarly, create a custom field to store `userId`, `Co-Id`, `JobId`, and `WorkId`.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle JD Edwards Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter <code>https://<JDEdwards-instance-name>.com</code>

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select No Security Policy .
Connectivity Agent	Select JDEAGENTGROUP . See Configure Oracle JD Edwards .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the integration flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
targetName	Specify the name of the view in Oracle JD Edwards from which the customer records must be retrieved
maxPageSize	Specify the maximum number of records per page
notificationMail	Specify the email address to which the error notifications are sent. Enter an email address of your choice
secret	Specify the password of your Oracle JD Edwards account
deviceName	Specify the machine name
username	Specify the username of your Oracle JD Edwards account

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

3. Run the recipe.

- a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
- b. On the Configure and run page, you can specify a value for the Select Date parameter and click **Run**.

The Select Date parameter stores the date of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Salesforce account records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, MM-DD-YYYY. To fetch all the account records from the beginning, set the value as zero.

You've now successfully submitted the integration for execution.

 Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Salesforce instance and check for the new accounts created.
 - a. On the Salesforce Setup page, click the **Accounts** tab.
 - b. On the Accounts Home page, select **All Accounts** in the **View** field. Click **Go** if necessary.

The list of accounts created are displayed.
 - c. To view an account record, click its name.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)
- [Using the Salesforce Adapter with Oracle Integration 3](#)

Import Customers and Credit Card Transactions from SAP ASE (Sybase) to Oracle Unity

Use this recipe to import customers and credit card transactions from SAP ASE (Sybase) to Oracle Unity.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports customer records and credit card transaction data from SAP ASE (Sybase) to Oracle Unity.

To use the recipe, you must install the recipe project and configure the connections and other resources within it. You can activate and run the integration flows manually or specify execution schedules for them. To fetch customer records from SAP ASE (Sybase) and import them to Oracle Unity, run the **Oracle Sybase Unity Customer Synch** integration flow. Similarly, to fetch credit card transaction records from SAP ASE (Sybase) and import them to Oracle Unity, run the **Oracle Sybase Unity CC Transaction Sync** integration flow.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- SAP ASE (Sybase)
- An account on SAP ASE Sybase with the Administrator role
- Oracle Unity
- An account on Oracle Unity with the Administrator role

Assumption

Ensure that the tables for customer profile data and transaction data are available in SAP ASE (Sybase) database.

Download [create_table_query_sybase.sql](#) for the sample table structure of the **CustomerProfile** table which stores customer profile data and **CustTransactions** table which stores transactions data.

Note

More fields can be added, or another table may be used; the SAP ASE (Sybase) endpoint in the recipe may then be modified as necessary.

Before You Install the Recipe

You must perform the following configuration tasks on your SAP ASE (Sybase) and Oracle Unity instances in order to successfully connect to these external systems using Oracle Integration and import customers and credit card transactions.

Configure SAP ASE (Sybase)

To successfully connect to SAP ASE (Sybase) using Oracle Integration and import customers and credit card transaction data, you must perform certain configurations on SAP ASE (Sybase).

Perform the prerequisite tasks to access the SAP ASE database:

- Ensure that you have write permissions on the database.
- Ensure that you have the required permissions to run stored procedures and packages and SQL statements against the SAP ASE database.
- Know the database hostname or IP address and the port number.
- Know the database name.
- Know the user name and password for connecting to the database.

- Install the connectivity agent. The connectivity agent is required to connect Oracle Integration with an SAP ASE (Sybase) on-premises database. See *Configure the Endpoint Access Type* in *Using the SAP ASE (Sybase) Adapter with Oracle Integration 3*.
- Download a Java Database Connectivity (JDBC) .jar file and place it in the third-party lib folder while configuring the connectivity agent. A JDBC driver enables a Java application to interact with a database. See [SAP ASE JDBC Driver](#).

Configure Oracle Unity

To configure Oracle Unity, see *Prerequisites for Creating a Connection* in *Using the Oracle Unity Adapter with Oracle Integration 3*.

Create custom object for credit card transactions by adding fields such as **Card Number**, **Remarks**, **Status**, **Card Type**, **TRXAMT**, **TRXDATE**, **Customer ID** and so on. For generic instructions to create custom objects in Oracle Unity, see [Creating custom data objects](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Sybase Connection

1. In the **Connections** section, click the connection name.
2. In the **Properties** section, enter the following details:

Field	Information to Enter
Host	Enter the host name of the database server.
Port	Enter the database server port number.
Database Name	Enter the database name.

3. In the **Security** section, enter the following details:

Field	Information to Enter
Security policy	Select Username Password Token .
Username	Enter the username. See Configure SAP ASE (Sybase) .
Password	Enter the password.

4. In the **Access type** section, select **Connectivity agent** and associate an agent group:
 - a. Click the **Associate agent group** button.
 - b. In the **Associate agent group** pane, select the agent group.
 - c. Click **Use**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Configure the Oracle Unity Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the URL to connect to Oracle Unity in the **Oracle Unity URL** field. For example: `https://XXXXXX.cxunity.ocs.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Access Token URI	Enter the authorization server that generates the access token.
Client ID	Enter the unique random string that matches the API console project.
Client Secret	Enter the unique random string that matches the API console project.
Username	Enter the Oracle Identity Cloud Service username of the OAuth token request.
Password	Enter the Oracle Identity Cloud Service password of the OAuth token request.
Scope	Enter the list of authorization permissions for the target application. For example, <code>read write</code> .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update integration properties of the integration flows **Oracle Sybase Unity CC Transaction Sync** and **Oracle Sybase Unity Customer Synch**. You can update the integration properties of the integration flows only after activating them.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.
 - i. **integration_prop_EmailTo**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
 - ii. **integration_prop_RetryCount**: You can specify the maximum number of retries if any invoke fails at run time. Default value is 2.

- iii. **integration_prop_Limit:** The pagination limit where you can specify the number of records per page. The default value specified is 999 and maximum value is 1000.
- c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
- 3. Run the recipe.
To fetch customer records from SAP ASE (Sybase) to Oracle Unity run the **Oracle Sybase Unity CC Transaction Sync** integration flow. Similarly, to fetch credit card transactions from SAP ASE (Sybase) to Oracle Unity run the **Oracle Sybase Unity Customer Synchronizer** integration flow.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

 Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

- 4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
- 5. In the Oracle Unity application, run the data warehouse job to copy and validate the records from temporary table to main data object. Once the data warehouse job is completed, check the records in customer object and custom transaction object for customer sync and transaction sync respectively.
- 6. Log in to your Oracle Unity instance and check for the new customers and credit card transaction records created.
 - a. In the landing page, click **Data Viewer**.
 - b. To view the customers, in the **Search or search for data object** field, type Customers, then select the relevant object from the suggestions that appear.
A list of customers created appears.
 - c. To view the credit card transactions, in the **Search or search for data object** field, type CustomTransactions, then select the relevant object from the suggestions that appear.
A list of custom transactions appears.

Related Documentation

- *Using the SAP ASE (Sybase) Adapter with Oracle Integration 3*
- *Using the Oracle Unity Adapter with Oracle Integration 3*

Import Documents from Oracle ERP Cloud to Adobe Sign

Use this recipe to import contract documents from Oracle ERP Cloud as agreements in Adobe Sign.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe sends the contract documents from Oracle ERP Cloud to Adobe Sign for electronic signing as per a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

When triggered, the integration flow pulls the contracts from the Oracle ERP Cloud instance based on their process ID and, in turn, extracts all the documents associated with each contract. Further, the recipe uploads the extracted documents into Adobe Sign as individual agreements. An email notification is sent to the individual agreement participants to sign the document.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- Adobe Sign
- An account on Adobe Sign with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Adobe Sign instances in order to successfully connect to these external systems using Oracle Integration and import contract documents.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Also, create the following global segment in the **ContractHeaderFlexfield** flexfield:

- **AdobeAgreementId**: Stores the contract agreement ID of the document in Adobe Sign.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user
3. Update the **ContractHeaderFlexfield** flexfield with the AdobeAgreementId global segment.
 - a. In the top-right corner of the Oracle ERP Cloud home page, click the profile image, then select **Setup and Maintenance**.
 - b. On the right, click **Task**, and select **Search**.
 - c. In the **Search** field, enter `Manage Descriptive Flexfields`, and then click **Search**.
 - d. From the search results, select **Manage Descriptive Flexfields**.
 - e. On the Manage Descriptive Flexfields page, search for the flexfield with Flexfield Code **ContractHeaderFlexfield**.
 - f. Create a global segment. See [Create Global Segments](#).
 - g. Under Create Segment, enter the Name of the segment as `AdobeAgreementId`. Also, set the API Name of the segment as `adobeagreementid` and choose the corresponding attribute.
 - h. Click **Save and Close**.
4. Deploy the descriptive flexfield. See [Deploy Descriptive Flexfields](#).

Configure Adobe Sign

To connect to Adobe Sign from Oracle Integration and send contract documents to it, you must create an OAuth application on your Adobe Sign instance and obtain access keys for the application. See *Prerequisites for Creating a Connection in Using the Adobe Sign Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Adobe Sign Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID of the OAuth application created in Adobe Sign. See Configure Adobe Sign .
Client Secret	Enter the client secret of the OAuth application created in Adobe Sign.
Scope	Enter user_read:account user_write:account user_login:account agreement_read:account agreement_write:account agreement_send:account.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

You'll now be redirected to the Adobe Sign login page.

6. Enter your Adobe Sign account credentials and click **Continue** to log in.

You are now displayed a page that prompts you to authorize the connection to the Adobe Sign application created previously.

7. On the Adobe Sign web page, click **ACCEPT**.

You're informed that access to Adobe Sign is allowed. You can now switch back to the Adobe Sign Connection window of Oracle Integration to test your connection.

8. In the Adobe Sign Connection window, click **Test** to ensure that your connection is successfully configured.

A message confirms if your test is successful.

9. To return to the project workspace, click **Go back** .

Configure the Oracle ERP REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORACLE-BRT-ERP_ADOBE_ESign_Constants** lookup table to map the fields related to contracts in Oracle ERP Cloud with agreements in Adobe Sign.

1. In the Lookups section, click the lookup name.
2. In the **Property_Name** column, enter the name of the Oracle ERP Cloud field you want to map.
3. In the **Property_Value** column, enter the name of the field in Adobe Sign to which the Oracle ERP Cloud has to be mapped or enter the default value for the property.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **Oracle ERP Adobe Document Create** integration flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
From_Address	Specify the email address from which the failure notification emails are sent. Enter an email address of your choice.
To_Address	Specify the email address to which the failure notification emails are sent. Enter an email address of your choice

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

3. Update property values for the **Oracle Adobe Document Create** integration flow.

- a. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
- b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
From_Address	Specify the email address from which the failure notification emails are sent. Enter an email address of your choice.
To_Address	Specify the email address to which the failure notification emails are sent. Enter an email address of your choice

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

4. Run the integration **Oracle ERP Adobe Document Create** to pull the contracts in the Oracle ERP Cloud. This integration, in turn, triggers the **Oracle Adobe Document Create** integration flow.

- a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the LastDocumentStartDate parameter. The LastDocumentStartDate parameter stores the date of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the records created after the date stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd. The default value of this parameter is 1999-01-01.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

6. Verify if an email notification is received by the individual agreement participants to sign the document imported from Oracle ERP Cloud.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Adobe Sign Adapter with Oracle Integration 3*

Import Documents from Salesforce to Adobe Sign

Use this recipe to import quote attachments from Salesforce to Adobe Sign.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that sends Salesforce quote attachments as agreements to Adobe Sign for electronic signing. Also, once the agreements are electronically signed, the recipe uploads the agreement back to the respective quote in Salesforce.

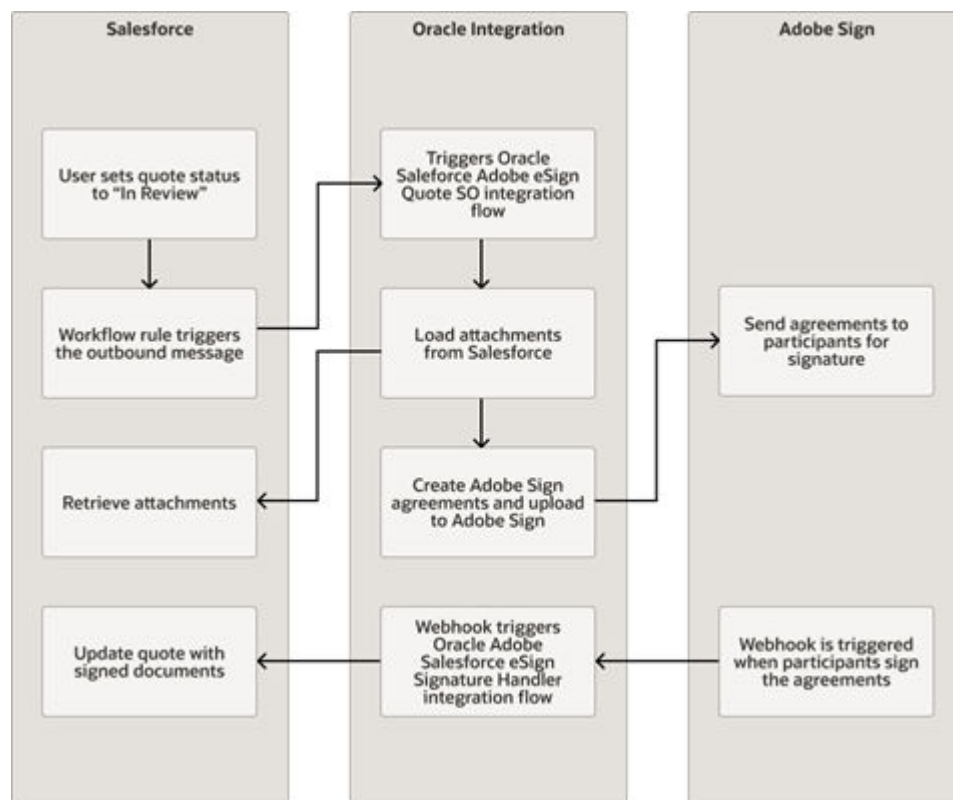
To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a Salesforce quote state is set to In-Review, the integration flow of the recipe is triggered, and it loads all the attachments in the quote. Further, the recipe uploads the quote attachments as individual agreements in Adobe Sign. An email notification is sent to the individual agreement participants to sign the document. Once the agreements are signed, the recipe then uploads the agreements back as attachments in the respective Salesforce quote.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Adobe Sign
- An account on Adobe Sign with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.



This recipe has the following integration flows:

- **Oracle Salesforce Adobe Sign Quote SO:** This integration flow is triggered when the status of a quote in Salesforce is set to In-Review. It retrieves the attachments from the quote and uploads the quote attachments as individual agreements in Adobe Sign.
- **Oracle Adobe Salesforce eSign Signature Handler:** This integration flow is triggered when the participants have signed the agreements. It updates the Salesforce quote with the signed documents.

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Adobe Sign instances in order to successfully connect to these external systems using Oracle Integration and import quote attachments.

Configure Salesforce

To access Salesforce from Oracle Integration, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the integration flow of the recipe, and create a workflow rule to send the outbound message when Salesforce quote attachments are signed.

Log in to your Salesforce instance as an **Administrator** and perform the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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. In the **Existing Profile** field, select **Standard User**.
 - ii. Enter a name for the new profile, for example, **API Enabled**, and click **Save**.

The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:

- i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field.
Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
 - e. Set a password and note the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lighting Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the **Quick Links** section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account.
Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
5. On your Salesforce instance, create an outbound message to trigger an integration flow of this recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.

- b. Click **New Outbound Message** in the resulting page.
- c. On the New Outbound Message page:
 - i. Select **Quote** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter the name for the outbound message as Oracle SFDC Adobe Esign Quote S0. 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 `https://<OIC_FQDN:OIC_SSL_PORT>/ic/api/integration/v2/flows/rest/project/ORCL-R-SFDC_ADOBE_ESIGN_S0/ORCL-R-SFDC_ESIGN_SIGN_HANDLER/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 **Quote 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 quote is created.
 - 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 **Quote** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:
 - Enter the rule's name. For example, Oracle SFDC Adobe Esign Quote S0.
 - 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 **criteria are met** option, and define the following rule criteria:
 - In the **Field** drop-down, select **Quote: Status**
 - In the **Operator** drop-down, select **equals**
 - In the **Value** drop-down, select **In Review**
 - 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, Oracle Adobe Sign Quote Message.
 - Click **Add** to move your selection to the **Selected Actions** box.
 - Click **Save**.
- v. Click **Done**.

The new workflow rule is created and displayed on your browser.

- d. Click **Activate** to activate the rule.
7. Create custom fields to hold document ID and status.
 - a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Customize**, then **Quote**, and then click **Fields**.
 - b. On the Quote Fields page, scroll to the **Quote Custom Fields & Relationships** section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter AdobeAgreementID 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.
 - 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 Quote Fields page, you can see the new field added under the **Quote Custom Fields & Relationships** section.

Similarly, create a custom field to store agreement status. Enter Oracle_Adobe_eSign_Agreement_Status as the field label.

Configure Adobe Sign

To access Adobe Sign using Oracle Integration and send agreements, you must perform certain configuration tasks on your Adobe Sign instance.

1. Create an OAuth application on your Adobe Sign instance and obtain access keys for the application. See Prerequisites for Creating a Connection in *Using the Adobe Sign Adapter with Oracle Integration 3*.
2. Create a webhook. See [Configure webhooks](#).
Enter the following details while configuring the webhook.

Field	Information to Enter
Name	Enter the name of the webhook.
Scope	Select Account .
URL	Enter https:// <username>:<password>@<OIC_Host_Name>/ic/api/ integration/v2/flows/rest/project/ORCL-R- SFDC_ADOBE_ESIGN_S0/ORCL-R- SFDC_ESIGN_SIGN_HANDLER/1.0/. Here, replace <username> and <password> with the username and password of your Adobe instance.
Events	Click the hamburger icon and select Agreement participant Complete .
Notifications Parameters	Select the following check boxes: Agreement Info , Agreement Document Info , Agreement Participant Info , Agreement Signed Document .

Click **Save**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Adobe REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Specify your Adobe URL. For example: <code>https://api.<domain_name>.adobesign.com/api/rest/v6</code> . Where <domain_name> indicates your domain such as in, us, eu, and so on.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Custom Three Legged Flow .
Authorization Request	The client application URL that the recipe must redirect you to when you provide consent. See Configure the REST Adapter to Consume a REST API protected with 3-legged OAuth token based authentication in <i>Using the REST Adapter with Oracle Integration 3</i>. For example: <code>https://secure.<domain_name>.adobesign.com/public/oauth/v2?response_type=code&client_id=<client_id_of_adobe_app>&redirect_uri=<OIC_Hostname>/icsapis/agent/oauth/callback&scope=user_login:self agreement_read:account agreement_write:account agreement_send:account</code>
Access Token Request	The access token request to use to fetch the access token. For example, <code>-X POST -H 'Content-Type: application/x-www-form-urlencoded' -d 'code=\${auth_code}&grant_type=authorization_code&redirect_uri=\${redirect_uri}&client_id=<client_id_of_adobe_app>&client_secret=<client_secret_of_adobe_app>' https://secure.<domain_name>.adobesign.com/oauth/v2/token</code>

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

You'll now be redirected to the Adobe Sign login page.

7. Enter your Adobe Sign account credentials and click **Continue** to log in.

You are now displayed a page that prompts you to authorize the connection to the Adobe Sign application created previously.

8. On the Adobe Sign web page, click **ACCEPT**.

You're informed that access to Adobe Sign is allowed. You can now switch back to the Adobe Sign Connection window of Oracle Integration to test your connection.

9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

10. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth 2.0 or Basic Authentication .

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORACLE-BRT-SFDC_ADOBE_ESign_Constants** lookup table to map the fields related to documents in Salesforce with agreements in Adobe Sign.

1. In the Lookups section, click the lookup name.
2. In the **Domain_Name_1** column, enter the name of the Salesforce field you want to map.
3. In the **Domain_Name_2** column, enter the name of the field in Adobe Sign to which the Salesforce field has to be mapped or enter the default value for the property.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe. To run the recipe, you must create and set the status of a quote in Salesforce as **In-Review**.
 - a. Create a quote in Salesforce. See [Create and Manage Quotes](#). Ensure that you upload relevant documents in the Notes and Attachments section of the quote.

- b. Mark the status of the quote as In-Review.

You've now triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify if an email notification is received by the individual agreement participants to sign the documents imported from Salesforce. Once the documents are signed by the participants, they are updated in the respective quote.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Adobe Sign Adapter with Oracle Integration 3*

Import Employee Records from an FTP Server to Oracle HCM Cloud

Use this recipe to import employee records from an FTP server to Oracle HCM Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports employee records from an FTP server to Oracle HCM Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When triggered, the recipe's integration flow picks a comma-separated values (CSV) file that contains employee records from a specified location in an FTP server. The imported CSV file is converted to NXSD format in Oracle Integration. The NXSD is used within Oracle Integration to generate a DAT file that is compatible with HCM Data Loader. The DAT file with employee records is then uploaded to Oracle HCM Cloud using HCM Data Loader.

Note that for successfully processing the employee records in Oracle Integration, the input file should be in a particular format. You can download a sample input file `employeeRecord.csv` from [Sample file](#).

System and Access Requirements

- Oracle Integration, Version 23.06.02 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- A secure FTP (SFTP) server
- An FTP client to access the SFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and Oracle HCM Cloud instance in order to connect to these external systems using Oracle Integration, and successfully import employee records.

Access Your FTP Server

Obtain an SFTP server, ensure that you're able to access it, and create required folders.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an input folder. Note the path of the folder.
The input CSV file with the employee records (employeeRecord.csv) is placed in this folder.
3. Create an attachment folder. Note the path of the folder.
Attachment files are placed in this folder.
4. Create an archive folder. Note the path of the folder.
The final zip file which will be uploaded to Oracle HCM Cloud is placed in this folder.

Configure Oracle HCM Cloud

To access Oracle HCM Cloud using Oracle Integration and import worker data, you must perform certain configurations on your Oracle HCM Cloud instance.

Log in to your Oracle HCM Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Note the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle HCM Cloud from Oracle Integration.
2. Enable the NXSD file generation feature.
This feature determines whether you can view the **Generate Mapping Files** parameter when you submit files for bulk data loading and whether mapping files are generated when you refresh business objects.
 - a. Navigate to **Setup and Maintenance**.
 - b. From the **Functional Areas** menu, select **HCM Data Loader**.
 - c. From the HCM Data Loader tasks, select **Configure HCM Data Loader**.
 - d. In the Configure HCM Data Loader window, under **Availability** category, select the **Generate Business Object Mapping Files** parameter.
 - e. Set the **Override** value of the parameter to **Yes**.
 - f. Click **Save and Close**.
3. Generate an NXSD file within HCM Data Loader.
To generate an NXSD file for a specific business object for the first time:
 - a. In the Oracle HCM Cloud navigator menu, under the Data Exchange section, click **HCM Data Loader**.

- b. In HCM Data Loader, click **View Business Objects**.
- c. Select the specific business object that you wish to generate an NXSD template for.
- d. Click **Refresh Object**.

An integration mapping file is generated and downloaded to your local system. Additionally, a copy of this file is also placed in Oracle WebCenter Content.

Note

If changes have occurred to the business object after the NXSD was downloaded, the above steps can be repeated to generate a new version.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your FTP server.
FTP Server Port	Enter the port to connect to your FTP server.
SFTP Connection	Select Yes or No based on your FTP server protocol.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **Oracle FTP HCM Import Document** integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the following integration properties with appropriate values.
 - **AttachmentsFolder**: Enter the path of the attachment folder that you created in your FTP server. See [Access Your FTP Server](#).
 - **ArchiveFolder**: Enter the path of the archive folder that you created in your FTP server.
 - **InputFolder**: Enter the path of the input folder that you created in your FTP server.
 - **FromAddress**: Enter the email address from which error notification emails have to be sent.
 - **ToAddress**: Enter the email address to which error notification emails have to be sent.
 - c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle FTP HCM Import Document** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Oracle HCM Cloud instance and check if the sample employee records have been imported.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*

Import Employee Records from Oracle HCM Cloud to Jira

Use this recipe to import employee records from Oracle HCM Cloud to Jira.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates corresponding users in Jira for employee records in Oracle HCM Cloud, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the employee records from the Oracle HCM Cloud instance and, in turn, creates corresponding user records in your Jira instance.

While running the recipe, you can configure the integration flow to read Oracle HCM Cloud employee records created after a specified time, thereby picking up only the new or required employee records in each run.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Atlassian
- An account on Atlassian 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

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Atlassian instances in order to successfully connect to these external systems using Oracle Integration and transfer employee records between them.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and read employee records, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Create an API Token in Your Atlassian Account

To access Jira from Oracle Integration and create user records, you'll require an API token associated with your Atlassian account.

To create an API token, see *Prerequisites to Use the Basic Authentication Security Policy in Using the Jira Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Jira Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Atlassian host name.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
User Name	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 .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table, and enter an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. In the **Value** column, enter an email ID of your choice.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the `lastRunDate` parameter. This parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Oracle HCM Cloud approved absence records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, MM-DD-YYYY. To fetch all the approved absence records from the beginning, set the value as zero.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Atlassian instance and check for the new users created.
 - a. On the Atlassian home page, select a Jira product, for example, Jira Work Management or Jira Software.
 - b. On the Jira product's page, click **People** in the tab bar, then click **Search people and teams**. Search for the required user records on this page.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the Jira Adapter with Oracle Integration 3*

Import Employee Records from Oracle HCM Cloud to Oracle E-Business Suite

Use this recipe to import employee records from Oracle HCM Cloud to Oracle E-Business Suite.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates corresponding employees in Oracle E-Business Suite for employee records in Oracle HCM Cloud, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the employee records from the Oracle HCM Cloud instance and, in turn, creates corresponding employee records in your Oracle E-Business Suite instance.

While running the recipe, you can configure the integration flow to read Oracle HCM Cloud employee records created after a specified time, thereby picking up only the new or required employee records in each run.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher

- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Oracle E-Business Suite
- An account on Oracle E-Business Suite with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Oracle E-Business Suite instances in order to successfully connect to these external systems using Oracle Integration and transfer employee records between them.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and read employee records, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure Oracle E-Business Suite

To access Oracle E-Business Suite from Oracle Integration, you must perform certain configurations on your Oracle E-Business Suite instance. In addition, you'll require a separate user account on Oracle E-Business Suite with necessary security grants.

Log in to your Oracle E-Business Suite instance as an **Administrator** and execute the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. 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 in Using the Oracle E-Business Suite Adapter with Oracle Integration 3*. Here, make sure you provide the security grants to the user account created for Oracle Integration.
3. Obtain the ID of the business group and organization in which the employees must be created.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://<hostname>/homePage/faces/FuseWelcome`.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle E-Business Suite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the URL of your Oracle E-Business Suite instance.
3. 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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Update Integration Properties

You have to update the integration property values so that the employee records are created in the right organization in Oracle E-Business Suite.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **OrgId**: This integration property holds the ID of the Oracle E-Business Suite organization in which the employees must be created. Enter the organization ID you obtained earlier. See [Configure Oracle E-Business Suite](#).

- b. BusinessGroupId:** This integration property holds the ID of the business group in Oracle E-Business Suite to which the new employees must belong. Enter the business group ID you obtained earlier. See [Configure Oracle E-Business Suite](#).
 - c. NotificationMail:** This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
- 3. Click **Submit**.**

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

- a. In the Integrations section of the project workspace, click **Actions** . . . on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the NewHireFetchTime parameter. This parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Oracle HCM Cloud employee records created after the date-time stamp stored as the parameter's Current Value are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the New Value field in the format, yyyy-MM-dd HH:mm. To fetch all the employee records from the beginning, set the value as zero.

- b.** On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle E-Business Suite instance and check for new employees.
 - a. Click the hamburger menu on the right pane. In the search bar, search and select **HRMS Manager**. Further, select **People** and choose **Enter and Maintain**.
 - b. Search and check for new employees.

Related Documentation

- Using the Oracle HCM Cloud Adapter with Oracle Integration
- Using the Oracle E-Business Suite Adapter with Oracle Integration

Import Employee Records from Oracle HCM Cloud to Salesforce

Use this recipe to import employee records from Oracle HCM Cloud to Salesforce.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates corresponding users in Salesforce for employee records in Oracle HCM Cloud, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the employee records from the Oracle HCM Cloud instance and, in turn, creates corresponding user records in your Salesforce instance.

While running the recipe, you can configure the integration flow to read Oracle HCM Cloud employee records created after a specified time, thereby picking up only the new or required employee records in each run.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and transfer employee records between them.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and read employee records, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and perform a few other configuration tasks. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure Salesforce

To access Salesforce from Oracle Integration and create user records, you must perform certain configurations 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.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
 - a. On the Salesforce Setup page:
 - i. Expand **Manage Users** under the Administer section in the left navigation pane.
 - ii. Click **Profiles**.
 - b. On the Profiles page, click **New Profile**.
 - c. On the resulting page:
 - i. Select **Standard User** in the **Existing Profile** field.
 - ii. Enter a name for the new profile, for example, API Enabled, and click **Save**.
The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user, for example, Integration User05.
 - ii. In the Email field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field.
Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce.

You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account.
Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.

4. Identify your Salesforce instance type. See *Identify the Instance Type of Your Salesforce Organization in Using the Salesforce Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Salesforce Username Password Policy .
Username	Enter the username 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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and enter an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. In the **OracleHCMDesignation** column, enter a designation available for Oracle HCM Cloud employee records. For example, Consultant or Senior Consultant.
3. In the **Salesforce_Profile_ID** and **Salesforce_Role_ID** columns, enter the IDs of the Salesforce profile and role with which the Salesforce user records must be created, for the specified HCM designation.
4. Against the **EmailTo** cell in the **Entity** column, enter an email address of your choice to receive error messages.
5. Click **Save**. If prompted, click **Save** again.
6. To return to the project workspace, click **Go back** .

Note

The lookup table contains some example values. Change them as necessary.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the LastRunTime parameter. The LastRunTime parameter stores the date and time of the

most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Oracle HCM Cloud employee records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the employee records from the beginning, set the value as zero.

- b.** On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Salesforce instance and check for the new users created.
 - a. On the Salesforce Setup page (in the Classic UI), expand **Manage Users** under the Administer section in the left navigation pane.
 - b. Click **Users**.

The list of users is displayed, which contains the user records created for employee records in Oracle HCM Cloud.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the Salesforce Adapter with Oracle Integration 3*

Import Employee Records from Oracle HCM Cloud to Zendesk

Use this recipe to import employee records from Oracle HCM Cloud to Zendesk.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates corresponding users in Zendesk for employee records in Oracle HCM Cloud, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the employee records from the Oracle HCM Cloud instance and, in turn, creates corresponding user records in your Zendesk instance.

While running the recipe, you can configure the integration flow to read Oracle HCM Cloud employee records created after a specified time, thereby picking up only the new or required employee records in each run.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Zendesk
- An account on Zendesk with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Zendesk instances in order to connect to these external systems using Oracle Integration and transfer employee records between them.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and read employee records, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure Zendesk

To access Zendesk from Oracle Integration and create or read user records, you'll need to create an OAuth client on your Zendesk instance. See *Create an OAuth Client in Using the Zendesk Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Connection Properties section, enter the Oracle HCM Cloud host name.
For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .

Field	Information to Enter
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Zendesk Connection

- In the Connections section, click the connection name.
- In the Connection Properties section, enter the Zendesk host name.
- In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID of the OAuth application created in Zendesk. See Configure Zendesk .
Client Secret	Enter the client secret of the OAuth application created in Zendesk.
Scope	Enter write.

- Click **Save**. If prompted, click **Save** again.
- Click **Provide Consent**.
- In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**. You'll be redirected to the Zendesk login page.
- Enter your Zendesk account credentials and click **Sign in**.
- After you see an access allowed message, switch back to the Zendesk Connection window of Oracle Integration to test your connection.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and enter an email ID to receive error messages.

- In the Lookups section, click the lookup name.
- In the **Value** column, enter an email ID of your choice.
- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).

2. Run the recipe.

- a. In the Integrations section of the project workspace, click **Actions** . . . on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the LastRunTime parameter.

The LastRunTime parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Oracle HCM Cloud employee records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the employee records from the beginning, set the value as zero.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Zendesk instance and check for the new users created.
 - a. On the Zendesk home page, click **Customers** in the left navigation pane.
 - b. Search for the required user records on the Customers page.

Related Documentation

- [Using the Oracle HCM Cloud Adapter with Oracle Integration 3](#)
- [Using the Zendesk Adapter with Oracle Integration 3](#)

Import Employee Vacation and Delegation Rules Report from Oracle HCM Cloud to Oracle Autonomous Transaction Processing

Use this recipe to import employee vacation and delegation rules report from Oracle HCM Cloud to Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)

- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe extracts employee vacation and delegation rules from Oracle Fusion Cloud and then stages these rules to ATP database for custom reporting and other needs. The employee vacation and delegation rules report is available as a scheduled process in Oracle Fusion Cloud. These rules are defined by users to reassign or delegate the workflow tasks assigned to them.

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe. A scheduled integration, using the Oracle ERP Cloud adapter, runs daily to extract rules based on parameters set in Oracle Integration. The extracted rules are saved as multiple files (XML, Excel, log) in Oracle Fusion Cloud. A callback integration triggered by Oracle Fusion Cloud upon process completion, retrieves the process ID and downloads the zip file (containing the extracted files) using the Oracle ERP Cloud adapter. The zip file is unzipped, and the XML data is extracted and loaded into the ATP database.

System and Access Requirements

- Oracle Integration 3
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your ATP and Oracle Fusion Cloud instances in order to successfully connect to these external systems using Oracle Integration and import employee vacation and delegation rules report.

1. Log in to your ATP instance as an **Administrator** and perform the following tasks.
 - a. Perform general configuration tasks. See *Prerequisites for Creating a Connection in Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.
 - b. Create necessary objects by running the following script.

Run the following script in ATP to create the necessary objects.

```

CREATE TABLE "ADMIN"."EMPLOYEEVACATIONRULE"
(
  "ID" NUMBER,
  "DES_USER_EMAIL" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "DES_USER_NAME" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "DES_USER_PERSON_NUMBER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "ORIG_USER_EMAIL" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "ORIG_USER_NAME" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "ORIG_USER_PERSON_NUMBER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "ACTION" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "START_DATE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "END_DATE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "CREATION_DATE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP"
) DEFAULT COLLATION "USING_NLS_COMP" ;

CREATE UNIQUE INDEX "ADMIN"."EMPLOYEEVACATIONRULE_PK" ON "ADMIN"."EMPLOYEEVACATIONRULE" ("ID")
;

ALTER TABLE "ADMIN"."EMPLOYEEVACATIONRULE" ADD CONSTRAINT "EMPLOYEEVACATIONRULE_PK" PRIMARY KEY ("ID")
USING INDEX ENABLE;

CREATE OR REPLACE EDITIONABLE TRIGGER "ADMIN"."EMPLOYEEVACRULE_TRG"
BEFORE INSERT ON EMPLOYEEVACATIONRULE
FOR EACH ROW
WHEN (new.Id IS NULL) BEGIN
:new.Id := EMPLOYEEVACRULE_SEQ.NEXTVAL;
END;

```

2. Log in to your Oracle Fusion Cloud instance and enable vacation period.
 - a. Click the bell icon on the top right and then choose **More Details**.
 - b. Click the Worklist.
 - c. From the opened Worklist, click the icon on the top right, and choose **Preferences**.
 - d. Select the **Enable vacation period** checkbox.
 - e. Choose the **Start Date** and **End Date**.
 - f. Select either **Reassign to** or **Delegation to** radio button.
 - g. Click the **Magnify** icon and search for the user to whom you want to reassign or delegate the tasks.
 - h. Click **Save**.
3. Create a user account for Oracle Integration on Oracle ERP Cloud. 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.

See Prerequisites for Creating a Connection in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See [Before You Install the Recipe](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Select the check box, then click Upload to upload the wallet file. See Before You Install the Recipe .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the recipe.

- a. In the Integrations section of the project workspace, click **Actions**  on the **HCM Employee Vacation and Delegation Rules Run** integration flow, then select **Run**.
- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

 Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify that the employee vacation and delegation rules have been written to the ATP database table.

Related Documentation

- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)
- [Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3](#)
- [Employee Vacation and Delegation Rules Report](#)

Import Employees from SAP SuccessFactors to Oracle E-Business Suite

Use this recipe to import employee records from SAP SuccessFactors to Oracle E-Business Suite.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports employee records from SAP SuccessFactors to Oracle E-Business Suite on a scheduled basis.

To use this recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

When triggered, the integration flow imports the employee records from SAP SuccessFactors to Oracle E-Business Suite. Basic employee data such as first name, last name, gender, date of birth, and so on, are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration Version 21.4.3.0.0 or higher
- SAP SuccessFactors.

- An account on SAP SuccessFactors with the Administrator role
- Oracle E-Business Suite
- An account on Oracle E-Business Suite with the Administrator role

Before You Install the Recipe

Perform the following configuration tasks on your SAP SuccessFactors and Oracle E-Business Suite instances to successfully connect to these external systems using Oracle Integration and transfer employee records.

Configure SAP SuccessFactors

To access SAP SuccessFactors and export employee records, you must perform certain configuration tasks on your SAP SuccessFactors instance.

For general configuration tasks, see Prerequisites for Creating a Connection in *Using the SAP SuccessFactors Adapter with Oracle Integration 3*.

In addition, you must create a custom field in the employee records of SAP SuccessFactors to store the corresponding Oracle E-Business Suite employee IDs.

To create the custom field:

1. Log in to the SAP SuccessFactors application.
2. In the **Search Tools** field, search for **Manage Business Configuration**.
3. In the **Admin Center** page, navigate to **Employee Central**, then **HRIS Elements**, then **Person Info**.
4. Under **HRIS Fields**, in the last row, select the **Identifier**, enter the **label** (example: Oracle EBS Employee ID), **Enable** the field, and select if the field is **Mandatory**, then click **Save**.
Click **details** to specify more attribute details such as character length, field criteria, trigger rule, and so on.

Configure Oracle E-Business Suite

To configure Oracle E-Business Suite, see Setup Tasks for Enabling the Oracle E-Business Suite Adapter in *Using the Oracle E-Business Suite Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the SAP SuccessFactors Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
SFAPI WSDL Location or Hostname	Enter only one of the following based on your requirement: - To execute the SOAP API: Enter the WSDL. For example: <code>https://host_name.successfactors.com/sfapi/v1/soap?wsdl</code> - The WSDL endpoint depends on where your SAP SuccessFactors instance is located. It can be in one of several data centers. Contact your SAP SuccessFactors representative if you are unsure of which data center to use. Access the WSDL by appending ?wsdl to the endpoint. - To execute the OData API: Enter the SAP SuccessFactors host name. For example: <code>https://host_name.successfactors.com</code> - The SAP SuccessFactors host name can easily be derived from the SAP SuccessFactors login URL.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select the security policy: • SuccessFactors Custom Policy: Select this security policy to execute the SOAP API. – In the Company ID field, enter the SAP SuccessFactors client instance. You received the company ID, username, and password when you subscribed to SAP SuccessFactors. – In the Username and Password fields, enter the username and password to connect to the SAP SuccessFactors instance. • SuccessFactors SAML Bearer Assertion Flow with Username Password Token Support: Select this security policy if you need to execute both the OData and SOAP APIs. – In the Company ID field, enter the SAP SuccessFactors client instance. You received the company ID, username, and password when you subscribed to SAP SuccessFactors. – In the API Key field, enter the API key that you obtained when creating a client application. See Create a Client Application, in <i>Using the SAP SuccessFactors Adapter with Oracle Integration 3</i>. – In the Private Key field, enter the private key that you obtained when creating a client application. See Create a Client Application, in <i>Using the SAP SuccessFactors Adapter with Oracle Integration 3</i>. – In the Username field, enter the username to connect to the SAP SuccessFactors instance. – (Optional) In the Password field, enter the password to connect to the SAP SuccessFactors instance. You can execute both the OData and SOAP APIs without providing the password because this is an optional field.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle E-Business Suite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the URL of your Oracle E-Business Suite instance.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
User Name	Enter a valid user name granted the privileges to access and execute the Oracle E-Business Suite APIs and services.
Password	Enter the password for the user name.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and enter parameter values corresponding to Oracle E-Business Suite.

1. In the Lookups section, click the lookup name.
2. Edit the lookup table and enter parameter values corresponding to Oracle E-Business Suite.

EBS Parameter	Description	Example Value
RespApplication	Application short name.	PER
SecurityGroup	Security group key	STANDARD
NLSLanguage	NLS language.	AMERICAN
Org_Id	Unique identifier for Oracle E-Business Suite.	204
P_BUSINESS_GROUP_ID	Unique identifier for Oracle E-Business Suite business group.	202

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration property values. For this integration flow, you can update these values only after activating it.
 - **EmailTo:** This integration property holds the email address to which runtime exception emails are sent. Enter an email address of your choice.
 - **RetryCount:** This integration property holds the number of retries to attempt if a record fails while importing employee records at run time. Enter a number of your choice.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed with the following parameter:

- **schedule_param_LastRun:** This parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the employees created in SAP SuccessFactors after the date-time stamp stored as the parameter's **Current Value** are processed by the integration flow in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your

choice (not beyond past three months) in the **New Value** field in the format yyyy-MM-dd HH:mm.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Check for the new employees created on your Oracle E-Business Suite instance.
 - a. Log in to the Oracle E-Business Suite instance.
 - b. In the top-left corner, click the menu icon, then navigate to **HRMS Management**, then **HRMS Manager**, then **People**, then **Maintain Using Templates**, then **Enter and Maintain**.

A Java file is downloaded to your system.
 - c. Open the Java file.
 - d. After the application is open, in the Find Person window, search for an employee record by name or number.

Related Documentation

- [Using the SAP SuccessFactors Adapter with Oracle Integration 3](#)
- [Using the Oracle E-Business Suite Adapter with Oracle Integration 3](#)

Import Files from an FTP Server to Oracle EPM Cloud

Use this recipe to import a file from an FTP server to Oracle Enterprise Performance Management (EPM) Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports a file from an FTP server and uploads it to Oracle EPM Cloud.

To use the recipe, you must install the recipe and configure connections and other resources within it. When you activate and run the recipe, the integration flow of the recipe is triggered. The integration fetches the file from the specified location on the FTP server, temporarily downloads the file in Oracle Integration, and subsequently uploads the file to Oracle EPM Cloud.

You can store the file to import on File Server, an embedded sFTP server within Oracle Integration. To use File Server, you must enable it for your instance. See [Enable File Server in Using File Server in Oracle Integration 3](#).

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Oracle EPM Cloud, 18.03 or higher
- A secure FTP (sFTP) server or File Server
- An FTP client to access the external sFTP server

Before You Install the Recipe

To successfully import files from an FTP server to Oracle EPM Cloud using Oracle Integration, you must perform the following tasks on your FTP server.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an input directory. Note down the path of the input directory.

In addition, review the steps required to upload files to Oracle EPM Cloud with Postman. See [Example: Using REST APIs to Upload with Postman](#) in *REST APIs for Oracle Fusion Cloud EPM*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the REST EPM Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Update Integration Properties

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **directory**: This is the folder location on the FTP server where the file is located. For example: /Data/Jan/records. See [Before You Install the Recipe](#). This is a required property.
 - b. **filename**: This is the file which has to be downloaded from the FTP server location. For example: dataFile.dat. This is a required property.
 - c. **stage_location**: This is the temporary location on Oracle Integration where the file would be downloaded. For example: tmp. This property is optional.
 - d. **epm_folder_path**: The folder in EPM cloud where the file has to be imported. For example: /inbox/Jan. This property is optional.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Log in to your FTP server and place a sample file in the input folder.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. In your Oracle EPM Cloud instance, check if the sample file has been imported into the specified folder.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using File Server in Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Import Files from an FTP Server to Oracle NetSuite

Use this recipe to import files from an FTP server to the NetSuite File Cabinet.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the 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

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.

Access Your FTP Server and Upload the Files

Obtain an SFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory (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 Oracle NetSuite

To access Oracle NetSuite using Oracle Integration and copy files, you must perform certain configuration tasks on your Oracle NetSuite instance.

To successfully connect to Oracle NetSuite, you must perform certain general configurations on your Oracle 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 Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See *Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3*.
2. Perform the TBA-related configuration tasks. See *Prerequisites for the Token-Based Authentication Security Policy in Using the Oracle NetSuite Adapter with Oracle Integration 3*.

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
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 Oracle 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Read-Write Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your SFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your SFTP server.
Password	Enter the password to connect to your SFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
WSDL URL	Enter the NetSuite WSDL URL; for example, <code>https://webservices.netsuite.com/wsd/</code> <code><NetSuite_application_version>/netsuite.wsdl</code> , where <code><NetSuite_application_version></code> is the version of the NetSuite application. For example, <code>v2015_1_0</code> . See <i>Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See <i>Create an Integration Record for Oracle Integration in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See <i>Create an Access Token for the User Account in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier. See <i>Make a Note of the NetSuite Account ID in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

Note

You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Integration Flow

Create a schedule for your integration to pick up files from the FTP server at a date, time, and frequency of your choosing.

- In the Integrations section, click **Actions**  on the integration flow, then select **Add Schedule**.
The page for defining the execution schedule is displayed.
- For this example, let's create a basic schedule, so leave the **Simple** radio button selected under the **Define recurrence** section.
- 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.

4. 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.
5. To return to the project workspace, click **Go back** .

To create advanced schedules, see *Define the Integration Schedule in Using Integrations in Oracle Integration 3*.

Activate and Run the Recipe

After you've configured the connections and the integration flow, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. 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. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Update Property Values**.
- b. In the Update Property Values pane, click 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 Oracle 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.
- c. Click **Submit**.

A message confirms 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 Oracle NetSuite instance.

- a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
- b. On the Configure and run page, click **Run**. Note that the **Ad hoc request** radio button is selected by default.

You've successfully submitted the integration for a test run.

4. Log in to your Oracle NetSuite instance and check for the imported file(s) in your File Cabinet folder.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle NetSuite Adapter with Oracle Integration 3*

Import Files from File Server to OCI Object Storage

Use this recipe to import files from File Server to OCI Object Storage.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to import files from File Server to OCI Object Storage using native actions. This solution is connection less, meaning no connections are used to connect to File Server and OCI Object Storage.

- **File Server:** A native action is used to connect to File Server without having to use the FTP adapter connection.
- **OCI Object Storage:** A native action is being used to connect to OCI Object Storage without having to use the REST adapter connection.

To use the recipe, you must install the recipe and configure the resources within it. Subsequently, you can activate and run the integration flow of the recipe. When triggered, the integration flow imports files from File Server into OCI Object Storage.

To use File Server which is an embedded sFTP server within Oracle Integration, you must enable it from your Oracle Integration instance. See *Enable File Server in Using File Server in Oracle Integration 3*.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- File Server
- OCI Object Storage
- An account on OCI Object Storage that has access to create objects

Before You Install the Recipe

You must perform the following configuration tasks on your File Server and OCI Object Storage instances in order to successfully import files from File Server to OCI Object Storage.

Configure File Server

You have to create the required directories in File Server.

1. Create a directory to import the input files from. Note down the path of the directory.

2. Create a directory to archive files that have been processed. Note down the path of the directory.
3. Create a directory where error files are to be placed.

Configure OCI Object Storage

Obtain access to OCI Object Storage and check the following:

1. If you don't already have a bucket where you can upload the imported files, create a bucket in a compartment.
2. Note down the bucket's location.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources.

Configure the Lookup Table

The recipe contains the **SFTPTToObjectStorageLKP** lookup table. Edit the lookup keys in the table as per your requirement.

1. In the Lookups section, click the lookup name.
2. Edit the lookup keys in the table.
 - **ToEmailAddress**: Specify the email address to which error notifications have to be sent.
 - **OCIErrorDirectory**: Specify the directory in File Server where files that errored out have to be sent.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you have configured the resources of the recipe, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **FileServer to OCI Storage Native file transfer** integration flow, then select **Run**.

The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

 - **Bucket Location**: This property holds the location of the bucket where the imported files have to be uploaded in OCI Object Storage.
 - **Bucket Name**: This property holds the name of the OCI Object Storage bucket where the imported files have to be uploaded.

- **SFTP_File_Path**: This property holds the path on File Server from which files have to be fetched.
 - **SFTP_Archive_File_Path**: This property holds the path on File Server where processed files are archived.
 - **File_Name_Pattern**: Specify the file name pattern.
- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In OCI Object Storage, check that the files have been imported to the specified bucket. In File Server, check if the processed files have been moved to the archived files directory .

Related Documentation

- [Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action](#)
- [Interact with Files in File Server](#)

Import Financial Journal Entries from Amazon S3 to Oracle ERP Cloud

Use this recipe to import financial journal entries from an Amazon S3 location to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe reads a file from an Amazon S3 location, does a simple transformation, and then imports the financial journal entries into Oracle ERP Cloud. The recipe uses File-Based Data Integration (FBDI).

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. The integration uses a scheduled orchestration pattern to download the file from the Amazon S3 location, process it, and import it into Oracle ERP Cloud. The import process records the results in a log file. To send this log file back to the Amazon S3 location, the integration uses an app-driven orchestration. The trigger to start an app-driven orchestration is based on an event happening. In this case, the log file is ready for export.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Amazon Web Services (AWS) with Amazon S3
- An account on AWS with Administrator role
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Amazon S3 and Oracle ERP Cloud instances in order to connect to these external systems using Oracle Integration, and successfully import financial journal entries from an Amazon S3 bucket to Oracle ERP Cloud.

Configure Amazon S3

Complete the following configuration tasks on your Amazon S3 instance to connect to it from Oracle Integration and import financial journal entries.

1. Obtain your AWS access keys. See *Prerequisites for Creating a Connection in Using the Amazon Simple Storage Service (S3) Adapter with Oracle Integration 3*.
2. Set up your Amazon S3 bucket where you want the resulting log file to be stored.
 - a. Log in to your AWS Management Console at <https://console.aws.amazon.com/>.
 - b. In the AWS Management Console, enter S3 in the search field at the top.
 - c. Select **S3 - Scalable Storage in the Cloud** from the search results.

The Amazon S3 page appears.
 - d. Select an existing bucket.
 - Scroll to the Buckets section. All available buckets are listed under it.
 - Select a bucket.
 - e. Alternatively, create and choose a new bucket.
 - i. Click **Create bucket**.
 - ii. In the Create bucket dialog, enter a unique name for the bucket.

Note some rules about naming an Amazon S3 bucket.

 - A bucket name should be unique across all Amazon S3 buckets.
 - Bucket names must be between 3 and 63 characters long.
 - Bucket names can consist only lowercase letters, numbers, dots (.), and hyphens (-).
 - You cannot write a bucket name as an IP Address, such as 192.168.0.1.
 - Bucket names must begin and end with a letter or number.
 - Bucket names should not contain two adjacent dots (..).
 - Bucket names should not end with -s3alias.
 - Bucket names should not start with xn--.

- f. In the **AWS Region** field, select a region from the drop-down list.
 - g. In the Object Ownership category, leave it as recommended. We use it for controlling the access of the files by specifying roles. If ACLs are disabled, the bucket owner automatically owns and has full control over every object in the bucket.
 - h. In Block Public Access settings for this bucket category, ensure that **BLOCK ALL PUBLIC ACCESS** is selected. If you want to host your static website in this bucket, you can change the settings later.
 - i. In the Bucket Versioning category, choose **Disabled**. Bucket versioning is helpful when you want to track any changes in the file made, intentionally or unintentionally. You can see the previous versions of a file, retrieve it, restore it or preserve it.
 - j. Leave other advanced settings as default.
 - k. Click **Create bucket**. The new bucket is listed under the Buckets section.
3. Note the AWS Region of the bucket.
 - a. Select the bucket.
 - b. Click the **Properties** tab, and note the **AWS Region** of the bucket.

Configure Oracle ERP Cloud

Complete the following configuration tasks on your Oracle ERP Cloud instance to connect to it from Oracle Integration, and import financial journal entries.

Perform the general configuration tasks for creating a connection. See Prerequisites for Creating a Connection in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Financial journal entries have to be in a particular format so that Oracle Enterprise Scheduler (ESS) Jobs can be run and the financial data can be finally stored in a log file. To enable this, you have to download the journal import template from Oracle ERP Cloud and prepare a journal import file based on it.

1. Download the journal import template (JournalImportTemplate.xlsx).
 - a. Navigate to [Oracle Fusion Cloud Applications Suite](#) in Oracle Help Center.
 - b. Under **Enterprise Resource Planning (ERP)**, click **Financials**.
 - c. In the left navigation pane of the Oracle Financials Get Started page, click **All Books**.
 - d. Under **Implementation**, open the *File-Based Data Import (FBDI) for Financials* guide.
 - e. Under the **General Ledger** section of the guide, click **Journal Import**.
 - f. In the **File Links** section, download the journal import template XLSM file (JournalImportTemplate.xlsx) and save it to your local machine.
2. Prepare the journal import data file.
 - a. Delete the sample data from the JournalImportTemplate.xlsx file. Keep only one record for your reference.
 - b. Populate the file with a few financial journal entry records.
 - c. Then, click and open the Instructions and CSV Generation sheet.
 - d. Click **Generate CSV File**.
 - e. Save the generated CSV file. It will be saved as a ZIP file on your local machine.

Upload the data file to the Amazon S3 bucket that you created in your Amazon S3 instance.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Amazon S3 Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the Amazon S3 connection URL: <code>https://s3.<your-bucket-region>.amazonaws.com/</code>

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select AWS Signature Version 4 .
Access Key	Enter the access key that you obtained. See Configure Amazon S3 .
Secret Key	Enter the secret key that you obtained.
AWS Region	Select the AWS region where you have created the Amazon S3 bucket.
Service Name	Select Amazon Simple Storage Service (Amazon S3) .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .

Field	Information to Enter
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORCL-BRT-S3_IMPORT_JOURNAL_SETTINGS** lookup table with appropriate values.

- In the Lookups section, click the lookup name.
- Enter appropriate values in the **Property_value** column for the following properties.

Property_name	Property_value
EmailErrorNotificationTo	Enter the email address where notifications will be sent in case of integration errors.
amazonS3ImportFileName	Enter the name of the financial journal file that have to be imported into Oracle ERP Cloud from the Amazon S3 bucket.
amazonS3BucketName	Enter the name of the Amazon S3 bucket.
amazonS3UploadFileName	Enter the name of the log file that has to be send back to the Amazon S3 location.

- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Update Integration Properties

The **Oracle AS3 ERP Journal Entry Import Callback** integration flow contains the **JobName** integration property that has to updated with appropriate value.

- In the Integrations section, click **Actions ...** on the integration flow, then select **Update property values**.
- In the Update property values panel, update the **JobName** integration property with appropriate value. The **JobName** integration property holds the name of the Oracle Enterprise Scheduler (ESS) job that you run in Oracle ERP Cloud.
- Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).

2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** . . . on the **Oracle AS3 ERP Journal Entry Import** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The **Oracle AS3 ERP Journal Entry Import** integration flow fetches the financial journal entries from Amazon S3 in the journal import file and imports them to Oracle ERP Cloud. In Oracle ERP Cloud, after the Oracle Enterprise Scheduler (ESS) job runs successfully, the imported financial journal entries are recorded in a log file.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

- c. Now, the **Oracle AS3 ERP Journal Entry Import Callback** integration flow gets triggered and it uploads the log file from Oracle ERP Cloud into the specified Amazon S3 bucket.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In the Amazon S3 instance, navigate to your Amazon S3 bucket and verify if the log file has been uploaded to it.

Related Documentation

- [Using the Amazon Simple Storage Service \(S3\) Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)
- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)

Import Financial Journal Entries from an FTP server to Oracle ERP Cloud

Use this recipe to import financial journal entries from an FTP Server location to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe reads a file from an FTP location, does a simple transformation, and then imports the financial journal entries into Oracle ERP Cloud. It uses File-Based Data Integration (FBDI). The integration flow can be invoked explicitly either as a web service or as a scheduled orchestration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe.

The integration flow of the recipe performs the following tasks:

- Reads the ledger-entries file from the FTP location.
- Unzip the file.
- Stages the file in Oracle Integration and read each entry.
- Transforms the data as required. For example, this integration changes the value in the currency column.
- Sends the file to Oracle ERP Cloud, configure a callback, and invoke the import process with the required parameters.
- After the import completes, Oracle Integration receives the result of the import from Oracle ERP Cloud. If the import completes successfully, then upload the log file to the FTP server. If the import fails, then send an email notification.

You can store the ledger-entries file on File Server, an embedded sFTP server within Oracle Integration. To use File Server, you must enable it for your Oracle Integration instance. See [Enable File Server](#).

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and Oracle ERP Cloud instances in order to connect to these external systems using Oracle Integration, and successfully import financial journal entries from an FTP server to Oracle ERP Cloud.

Access Your FTP Server

Obtain an sFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an output directory (ftp-erp-transfer) on the server where the output log files from Oracle ERP Cloud must be uploaded . Note down the path of this directory.

Configure Oracle ERP Cloud

Complete the following configuration tasks on your Oracle ERP Cloud instance to connect to it from Oracle Integration, and import financial journal entries.

Perform the general configuration tasks for creating a connection. See [Prerequisites for Creating a Connection](#).

Financial journal entries have to be in a particular format so that Oracle Enterprise Scheduler (ESS) Jobs can be run and the financial data can be finally stored in a log file. To enable this, you have to download the journal import template from Oracle ERP Cloud and prepare a journal import file based on it.

1. Download the journal import template (JournalImportTemplate.xlsx).
 - a. Navigate to [Oracle Fusion Cloud Applications Suite](#) in Oracle Help Center.
 - b. Under **Enterprise Resource Planning (ERP)**, click **Financials**.
 - c. In the left navigation pane of the Oracle Financials Get Started page, click **All Books**.
 - d. Under **Implementation**, open the *File-Based Data Import (FBDI) for Financials* guide.
 - e. Under the **General Ledger** section of the guide, click **Journal Import**.
 - f. In the **File Links** section, download the journal import template XLSM file (JournalImportTemplate.xlsx) and save it to your local machine.
2. Prepare the journal import data file.
 - a. Delete the sample data from the JournalImportTemplate.xlsx file. Keep only one record for your reference.
 - b. Populate the file with a few financial journal entry records.
 - c. Then, click and open the Instructions and CSV Generation sheet.
 - d. Click **Generate CSV File**.
 - e. Save the generated CSV file. It will be saved as a ZIP file on your local machine.

Upload the data file to the input folder in your FTP server.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure Oracle ERP Cloud

Complete the following configuration tasks on your Oracle ERP Cloud instance to connect to it from Oracle Integration, and import financial journal entries.

Perform the general configuration tasks for creating a connection. See [Prerequisites for Creating a Connection](#).

Financial journal entries have to be in a particular format so that Oracle Enterprise Scheduler (ESS) Jobs can be run and the financial data can be finally stored in a log file. To enable this, you have to download the journal import template from Oracle ERP Cloud and prepare a journal import file based on it.

- Download the journal import template (JournalimportTemplate.xlsm).
 - Navigate to [Oracle Fusion Cloud Applications Suite](#) in Oracle Help Center.
 - Under **Enterprise Resource Planning (ERP)**, click **Financials**.
 - In the left navigation pane of the Oracle Financials Get Started page, click **All Books**.
 - Under **Implementation**, open the *File-Based Data Import (FBDI) for Financials* guide.
 - Under the **General Ledger** section of the guide, click **Journal Import**.
 - In the **File Links** section, download the journal import template XLSM file (JournalimportTemplate.xlsm) and save it to your local machine.
- Prepare the journal import data file.
 - Delete the sample data from the JournalimportTemplate.xlsm file. Keep only one record for your reference.
 - Populate the file with a few financial journal entry records.
 - Then, click and open the Instructions and CSV Generation sheet.
 - Click **Generate CSV File**.
 - Save the generated CSV file. It will be saved as a ZIP file on your local machine.

Upload the data file to the input folder in your FTP server.

Configure the Lookup Table

Edit the **ORCL-BRT-ERP_IMPORTJOURNAL** lookup table and enter appropriate values.

- In the Lookups section, click the lookup name.
- Enter appropriate values in the **Property_value** column for the following properties.

Property_name	Property_value
outputDirectoryOnFTP	Enter the path of the output directory that you created on the FTP server. See Access Your FTP Server .
emailTo	Enter the email address where notifications errors have to be sent.
uploadFileName	Enter the name of the log file with the financial journal entries.

- Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Update Integration Properties

The **Oracle ERP FTP Import Journal Callback** integration flow contains the **JobName** integration property that has to be updated with appropriate value.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the **JobName** integration property with appropriate value. The **JobName** integration property holds the name of the Oracle Enterprise Scheduler (ESS) job that you run in Oracle ERP Cloud.
3. Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle FTP ERP Import Journal** integration flow, then select **Run**.

You've now successfully submitted the integration for execution. The **Oracle FTP ERP Import Journal** integration flow fetches the financial journal entries from the FTP server in the journal import file and imports them to Oracle ERP Cloud. In Oracle ERP Cloud, after the Oracle Enterprise Scheduler (ESS) job runs successfully, the imported financial journal entries are recorded in a log file.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

- c. Now, the **Oracle ERP FTP Import Journal Callback** integration flow gets triggered and it uploads the log file from Oracle ERP Cloud into the specified output folder in the FTP server.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
 4. Log in to the FTP server and check if the log file with the imported financial journal entries have been uploaded to the output folder.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using File Server in Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Import Leads from SugarCRM to Oracle ERP Cloud

Use this recipe to import leads from SugarCRM to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports leads from SugarCRM to Oracle ERP Cloud according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the lead records from the SugarCRM instance and, in turn, creates corresponding records in your Oracle ERP Cloud instance. Basic data such as name, phone, address, city, State, Postal Code, and so on are synchronized between the SugarCRM and Oracle ERP Cloud platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- SugarCRM
- An account on SugarCRM with the Administrator role
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SugarCRM and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration.

Configure SugarCRM

To configure SugarCRM:

1. Ensure that you satisfy the Prerequisites for Creating a Connection in *Using the SugarCRM Adapter with Oracle Integration 3*.
2. Create a custom field in SugarCRM to store the IDs of the leads created in Oracle Integration.

To create a custom field, execute the following tasks:

1. Log in to your SugarCRM instance using the SugarCRM admin credentials.
2. On the landing page, in the upper-right corner, click your profile icon, and select **Admin**.

3. On the Administration page, click **Studio** under **Developer Tools**.
4. On the Studio page, choose **Leads**.
5. On the Edit Leads page, click **Fields**.
6. On the Edit Fields page, click **Add Field** and complete the following:
 - Select the data type of the field.
 - Enter the field name.
 - Specify the character length.
7. Click **Save**.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle SugarCRM Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Site URL	Enter the SugarCRM instance URL. For example: <code>https://sg-your_instance_name.demo.sugarcrm.com</code> .
REST Version	Enter the supported SugarCRM REST version. For example: <code>v11_8</code> .

3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID that you obtained earlier. See Configure SugarCRM .
Client Secret	Enter the client secret that you obtained earlier.
Username	Enter the SugarCRM instance username.
Password	Enter the SugarCRM instance password.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the recipe.
 - In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - On the Configure and run page, you can specify a value for the LastRunTime parameter, and click **Run**.

The LastRunTime parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each

successful run, and only the SugarCRM lead records created 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. To fetch all the supplier records from the beginning, set the value as zero. You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle ERP Cloud instance and check for the new leads created.
 - a. On the home page, click **Menu**.
 - b. Under Sales, click **Leads**.

The list of leads is displayed, which contains the records imported from SugarCRM.

Related Documentation

- [Using the SugarCRM Adapter with Oracle Integration 3](#)
- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)

Import Marketo Leads to Oracle Autonomous Data Warehouse

Use this recipe to import campaign leads from Marketo into Oracle Autonomous Data Warehouse (ADW).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe bulk imports campaign leads created in Marketo and writes them into Oracle Autonomous Data Warehouse 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 and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow manually or specify an execution schedule for it. When triggered, the integration flow 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 Autonomous Data Warehouse.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher

- Marketo
- Oracle Autonomous Data Warehouse, 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 Autonomous Data Warehouse with the Administrator role
- An Oracle Autonomous Data Warehouse table with write access

Before You Install the Recipe

You must perform the following configuration tasks on your Marketo and Oracle Autonomous Data Warehouse instances in order to connect to these external systems using Oracle Integration and successfully import leads from Marketo into 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 3*.

1. Create an API only user role.
2. Create a customer service.
3. Obtain the Client ID and Client Secret.
4. Obtain the Munchkin ID.

Configure Oracle Autonomous Data Warehouse

Complete configuration tasks in your Oracle Autonomous Data Warehouse 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 Autonomous Data Warehouse 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 3*.

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.

2. Get the Autonomous Data Warehouse 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 Autonomous Data Warehouse 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 Autonomous Data Warehouse connection security from Oracle Integration.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Marketo Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:
 - In the **Munchkin ID** field, enter the munchkin ID that you obtained earlier. See [Configure Marketo](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID obtained earlier. See Configure Marketo .
Client Secret	Enter the client secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle ADW Connection

Configure the connection details for the Oracle Autonomous Data Warehouse connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

In the **Service Name** field, enter the service name of your Oracle Autonomous Data Warehouse instance. See [Configure Oracle Autonomous Data Warehouse](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL from the drop-down list.
Wallet	Click Upload to upload the credential wallet file that you obtained earlier. See Configure Oracle Autonomous Data Warehouse .
Wallet Password	Enter the password for your credential wallet file.
Database Service Username	Enter your Oracle Autonomous Data Warehouse account username.
Database Service Password	Enter your Oracle Autonomous Data Warehouse account password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

Note

Before activating and running the recipe, ensure that there are some campaign leads in the Marketo instance.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the `LeadStartAt` parameter. In the **New Value** field, enter the date from which you want to import campaign leads from Marketo, in the format `YYYY-MM-DDThh:mm:ssZ`. For example, `2021-09-27T00:00:00Z`.
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.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle Autonomous Data Warehouse instance and check if the campaign leads from Marketo have been imported into the Autonomous Data Warehouse database table.

Related Documentation

- [Using the Marketo Adapter with Oracle Integration 3](#)
- [Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3](#)

Import Offer Letters from Oracle HCM Cloud to Adobe Sign

Use this recipe to import offer letters from Oracle HCM Cloud to Adobe Sign.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)

- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports job offer letters as agreements into Adobe Sign for electronic signing, according to a schedule specified in Oracle Integration. Also, once the agreements are electronically signed, the recipe uploads the offer letters back to the respective candidate's record in Oracle HCM Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the recipe retrieves the candidate information and the offer template. Further, the recipe generates offer letters and uploads them as individual agreements in Adobe Sign.

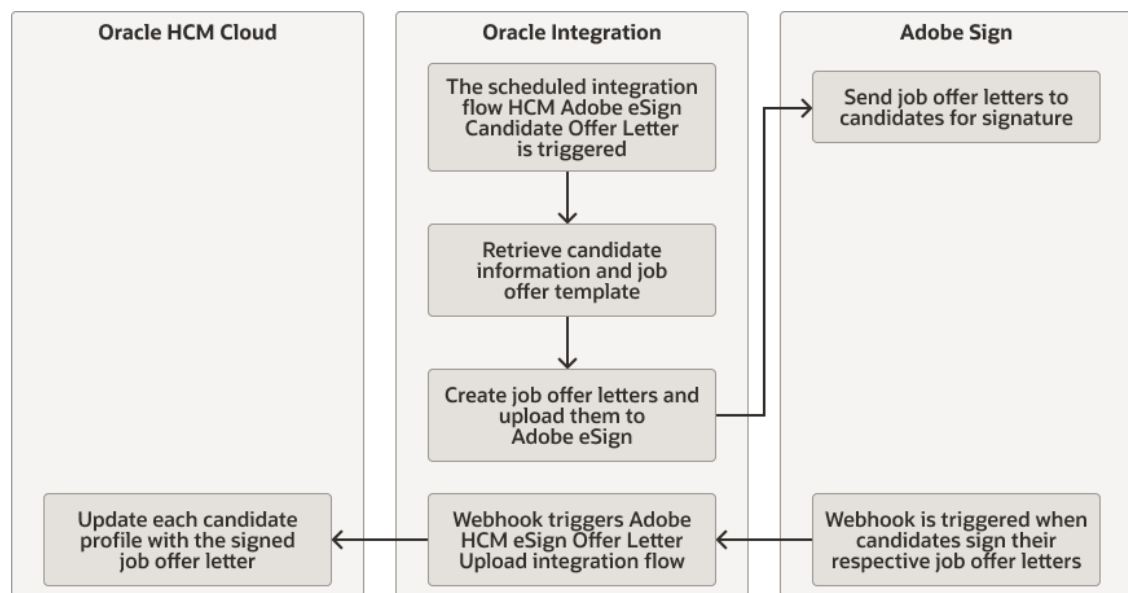
An email notification is sent to the individual candidates to sign the document. Once the offer letters are digitally signed, the recipe then uploads each offer letter to the corresponding candidate's record in Oracle HCM Cloud.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- Adobe Sign
- An account on Adobe Sign with the Administrator role
- Adobe Document Service
- An account on Adobe Document Service with the Administrator role
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Recipe Schema

This section provides an architectural overview of the recipe.



This recipe has the following integration flows:

- **HCM Adobe eSign Candidate Offer Letter:** This scheduled integration flow, when triggered, retrieves the candidate information and the offer template document. Further, using the retrieved details, it generates the offer letters and uploads them to Adobe Sign.
- **Adobe HCM eSign Offer Letter Upload:** This integration flow is triggered when the candidates have signed the offer letters. It updates the corresponding candidate records in Oracle HCM Cloud with the signed offer letters.

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Adobe Sign instances in order to successfully connect to these external systems using Oracle Integration and import offer letters.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and import offer letters, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure Adobe Sign

To access Adobe Sign using Oracle Integration and send agreements, you must perform certain configuration tasks on your Adobe Sign instance.

1. Create an OAuth application on your Adobe Sign instance and obtain access keys for the application. See *Prerequisites for Creating a Connection in Using the Adobe Sign Adapter with Oracle Integration 3*.
2. Create a webhook. See [Configure webhooks](#). Enter the following details while configuring the webhook.

Field	Information to Enter
Name	Enter the name of the webhook.
Scope	Select Account .
URL	Enter https:// <username>:<password>@<OIC_Host_Name>/ic/api/ integration/v2/flows/rest/project/ORCL-R- HCM_ADOBE_OFFER_LETTER/ ADOBE_HCM_ESIGN_OFFER_LETTE_UPL0/1.0. Here, replace <username> and <password> with the username and password of your Adobe instance.
Events	Click the hamburger icon and select Agreement participant Complete .
Notifications Parameters	Select the following check boxes: Agreement Info , Agreement Document Info , Agreement Participant Info , Agreement Signed Document .

Click **Save**.

Configure Adobe PDF Service Authentication

To access Adobe PDF Service using Oracle Integration, you must perform certain configuration tasks on your Adobe PDF Service instance.

1. Create a project in Adobe developer console. See [Access Adobe Developer Console](#).
2. Select and add **PDF Service API** to your project.
 - a. In your project's home page, click **Add to Project**, and select **API**.
 - b. From the list of APIs, select **PDF Service API**, and click **Next**.
 - c. Select **Service Account (JWT)** as your authentication method and click save.
This downloads a zip file named **config.zip** with the private key. Save the file.
3. Generate a signed JSON Web Token (JWT).
 - a. Sign in to your [Adobe developer console](#).
 - b. Select your project.
 - c. Click **Service Account (JWT)** and save the client ID and client secret. Further, navigate to the Generate JWT tab.
 - d. Paste your private key and click **Generate Token**.
 - e. Save the JWT.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Adobe eSign REST Connection

1. In the Connections section, click the connection name.

- In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Specify your Adobe URL. For example: <code>https://api.<domain_name>.adobesign.com/api/rest/v6</code> . Where <code><domain_name></code> indicates your domain such as <code>in</code> , <code>us</code> , <code>eu</code> , and so on.

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Custom Three Legged Flow .
Authorization Request	The client application URL that the recipe must redirect you to when you provide consent. See <i>Configure the REST Adapter to Consume a REST API protected with 3-legged OAuth token based authentication in Using the REST Adapter with Oracle Integration 3</i> . For example: <code>https://secure.<domain_name>.adobesign.com/public/oauth/v2?response_type=code&client_id=<client_id_of_adobe_app>&redirect_uri=<OIC_Hostname>/icsapis/agent/oauth/callback&scope=user_login:self agreement_read:account agreement_write:account agreement_send:account</code>
Access Token Request	The access token request to use to fetch the access token. For example, <code>-X POST -H 'Content-Type: application/x-www-form-urlencoded' -d 'code=\${auth_code}&grant_type=authorization_code&redirect_uri=\${redirect_uri}&client_id=<client_id_of_adobe_app>&client_secret=<client_secret_of_adobe_app>' https://secure.<domain_name>.adobesign.com/oauth/v2/token</code>

- Click **Save**. If prompted, click **Save** again.
- Click **Provide Consent**.
- 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 Adobe Sign login page.
- Enter your Adobe Sign account credentials and click **Continue** to log in.

You are now displayed a page that prompts you to authorize the connection to the Adobe Sign application created previously.
- On the Adobe Sign web page, click **ACCEPT**.

You're informed that access to Adobe Sign is allowed. You can now switch back to the Adobe Sign Connection window of Oracle Integration to test your connection.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

10. To return to the project workspace, click **Go back** .

Configure the Oracle REST Trigger Connection

1. In the Connections section, click the connection name.
 2. In the Security section, enter the following details:
- | Field | Information to Enter |
|-----------------|--|
| Security Policy | Select OAuth 2.0 or Basic Authentication . |
3. Click **Save**. If prompted, click **Save** again.
 4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the Oracle FTP Connection

Use the following steps to configure the **Oracle FTP Read Write Connection**.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:
 - a. **FTP Server Host Address**: Enter the host address of your SFTP server.
 - b. **FTP Server Port**: Enter 21.
 - c. **SFTP Connection**: specify the value as Yes.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your SFTP server.
Password	Enter the password to connect to your SFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle HCM Cloud Connection

Use the following steps to configure the **Oracle HCM Cloud Connection** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Adobe IDM Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Specify your connection URL. For example: <code>https://<instance_name>.adobelogin.com</code> .

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the Oracle Adobe REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Specify your connection URL. For example: <code>https://pdf-services.adobe.io</code>

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **HCM-ADOBE-CONFIG-LOOKUP** lookup table to map the fields related to offer letters in Adobe Sign.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

Key	Value
client_id	Enter the client ID obtained while configuring Adobe PDF service authentication. See Configure Adobe PDF Service Authentication .
client_secret	Enter the client secret obtained while configuring Adobe PDF service authentication.
jwt_token	Enter the JWT token obtained while configuring Adobe PDF Service Authentication.
SignatureType	Enter ESIGN.
RecipientRole	Enter SIGNER.
SigningOrder	Enter 1.
CompanyName	Enter the name of the organization releasing the offer letters
CompanyContactNumber	Enter the contact number of the organization

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **HCM Adobe eSign Candidate Offer Letter** integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
wait_time	Specify the number of seconds the integration must wait for the offer letter to be generated in Adobe.
templateName	Specify the name of offer letter document template in the FTP directory.
directory	Specify the path of the directory where the offer letter document template is stored.

3. Run the **HCM Adobe eSign Candidate Offer Letter** integration flow.

- a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Verify if an email notification is received by the individual agreement participants to sign the documents imported from Oracle HCM Cloud. Once the documents are signed by the participants, they are uploaded in the respective candidate's record in Oracle HCM Cloud.

Related Documentation

- [Using the Oracle HCM Cloud Adapter with Oracle Integration 3](#)
- [Using the Adobe Sign Adapter with Oracle Integration 3](#)

Import Oracle Commerce Cloud Orders to Oracle Intelligent Advisor Interviews

Use this recipe to import Oracle Commerce Cloud orders to Oracle Intelligent Advisor interviews.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports orders from Oracle Commerce Cloud and loads them to an interview form in Oracle Intelligent Advisor. Additionally, it also loads the corresponding product details of a selected order and displays them in the interview form.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a customer launches an interview form in Oracle Intelligent Advisor and enters his profile id in the Customer page, the first integration flow (**Oracle IA CX Commerce Get Orders**) is triggered in Oracle Integration. The integration checks if the entered profile id exists in Oracle Commerce Cloud. If the profile id exists, it fetches the orders created against it from Oracle Commerce Cloud, and loads them into the Oracle Intelligent Advisor interview's Customer Orders page. Subsequently when the customer selects an order in the Customer Orders page, the second integration flow (**Oracle IA CX Commerce Get Order Details**) is triggered in Oracle Integration. The integration fetches the corresponding product details of the selected order from Oracle Commerce Cloud and displays them in the Oracle Intelligent Advisor interview's Customer Products page.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- Oracle Commerce Cloud
- An account on Oracle Commerce Cloud with the Administrator role
- Oracle Intelligent Advisor
- An account on Oracle Intelligent Advisor with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Commerce Cloud and Oracle Intelligent Advisor instances in order to successfully connect to these external systems using Oracle Integration and import Oracle Commerce Cloud orders to Oracle Intelligent Advisor interviews.

Configure Oracle Commerce Cloud

To configure Oracle Commerce Cloud, see *Prerequisites for Creating a Connection in Using the Oracle Commerce Cloud Adapter with Oracle Integration 3*.

Configure Oracle Intelligent Advisor

To access Oracle Intelligent Advisor using Oracle Integration and load Oracle Commerce Cloud orders in Oracle Intelligent Advisor interview forms, you must perform certain configurations on your Oracle Intelligent Advisor instance.

1. Import the project into Policy Modeling, the interview designer.
Download the sample project file to import - [eCommerce Orders Management.zip](#).
2. Deploy the project into Oracle Intelligent Advisor Hub.
3. [Create a connection to Oracle Integration from Intelligent Advisor Hub](#) using OAuth.
4. [Create a confidential application to connect to Oracle Integration](#).

For more information, see [Use the Intelligent Advisor Adapter for Interviews](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Commerce Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the **Connection base URL** to connect to Oracle Commerce Cloud. For example: `http://<host:port>/ccadmin/v1`.
3. In the Security section, enter the **Security Token**. See [Configure Oracle Commerce Cloud](#).

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle Intelligent Advisor Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the **Intelligent Advisor Hub URL**. For example: `https://<your_site_interface>/opa-hub`.
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Enter the access token URI. For example: <code>https://<your_site_interface>/opa-hub/api/auth</code>
Client Id	Enter the name of the API Client with the Manager role in Intelligent Advisor Hub. See Configure Oracle Intelligent Advisor .
Client Secret	Enter the password for the API Client with the Manager role in Intelligent Advisor Hub.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Update Integration Property

Update the **FromDate** integration property for the **Oracle IA CX Commerce Get Orders** integration flow.

- In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
- In the Update property values panel, enter an appropriate value in the **New Value** field for the **FromDate** integration property. For example: `2020-01-24T00:00:44.000Z`.
- Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the recipe.

- a. Log in to your Oracle Intelligent Advisor instance and launch an interview.
- b. Enter the customer profile Id and click **Next**.

This triggers the **Oracle IA CX Commerce Get Orders** integration flow which loads the orders for the chosen profile Id into the Customer Orders page of the interview form.

- c. In the Customer Orders page, select an order from the order list.

The **Oracle IA CX Commerce Get Order Details** integration flow is triggered and it loads the product details of the order selected into the Products page of the interview form.

You've successfully imported orders from Oracle Commerce Cloud to an Oracle Intelligent Advisor interview.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

Related Documentation

- *Using the Oracle Commerce Cloud Adapter with Oracle Integration 3*
- *Using the Oracle Intelligent Advisor Adapter with Oracle Integration 3*

Import Orders from Oracle ERP Cloud to Oracle Autonomous Data Warehouse

Use this recipe to import orders from Oracle ERP Cloud to Oracle Autonomous Data Warehouse (ADW).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports the order details of a new order created in Oracle ERP Cloud (Order Management Module) to a database table in Oracle Autonomous Data Warehouse (ADW).

To use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe contains two integration flows, one for receiving the orders from Oracle ERP Cloud and then updating the same into an Apache Kafka topic, and another for writing the orders from the Apache Kafka topic into the ADW database table. When an order is created in Oracle ERP Cloud, the first integration (**Oracle ERP Kafka Order Sync Producer**) is triggered. The integration flow receives the order details from Oracle ERP Cloud and writes it into an Apache Kafka topic. This in turn triggers the second integration (**Oracle Kafka ADW Order Sync Consumer**) which consumes the order from the Apache Kafka topic and writes it to the ADW database table, thereby importing the newly created order from Oracle ERP Cloud.

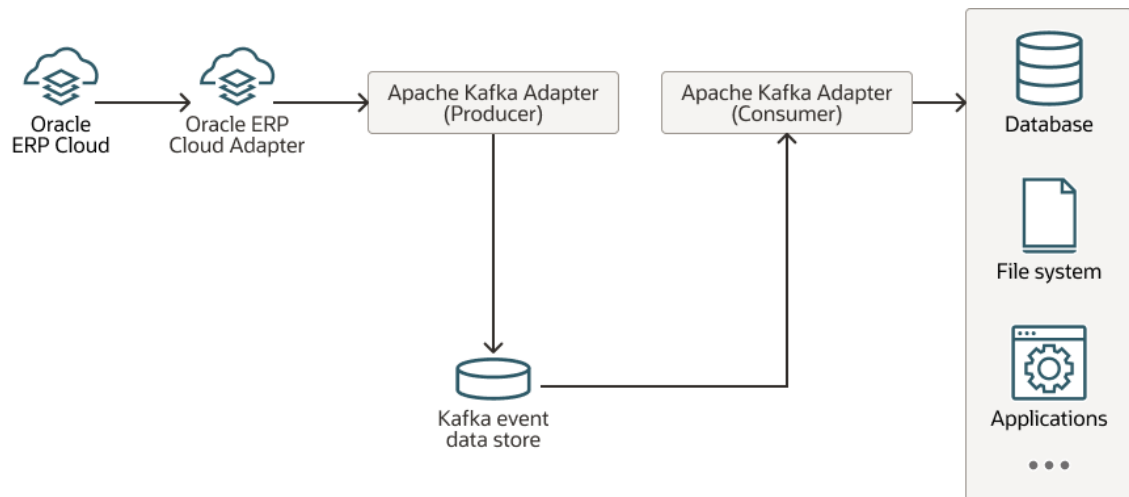
System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- Oracle ERP Cloud

- Oracle Autonomous Data Warehouse
- Apache Kafka
- An account on Oracle ERP Cloud with the Administrator and Procurement Manager roles
- An account on Oracle Autonomous Data Warehouse with the Administrator role
- An Oracle Autonomous Data Warehouse database table with write access
- An account on Apache Kafka with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.



An event is raised in Oracle ERP Cloud when an order is created. 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 order details to an Apache Kafka topic. This in turn triggers the second integration flow which uses the Apache Kafka Adapter (Consumer) and Oracle Autonomous Data Warehouse Adapter to write the order details from the Apache Kafka topic to an ADW database table.

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud, Apache Kafka, and Oracle Autonomous Data Warehouse instances in order to successfully connect to these external systems using Oracle Integration and import orders.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles/privileges to the user account. See *Assign Required Roles to an Integration User* in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

- Integration Specialist
- AttachmentsUser
- FND_MANAGE_CATALOG_SERVICE_PRIV

For more details, see Prerequisites for Creating a Connection in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.

Configure Apache Kafka

Complete the following configuration tasks on your Apache Kafka instance to successfully connect to it from Oracle Integration and import orders.

1. Get the Apache Kafka Cluster Bootstrap Server URL. See Prerequisites for Creating a Connection in *Using the Apache Kafka Adapter with Oracle Integration 3*.

2. 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 topics in *Using Integrations in Oracle Integration 3*:

- System Requirements
- Connectivity Agent Restrictions

To install and configure the connectivity agent:

- a. Create an agent group.
 - b. Download and install the connectivity agent.
3. Configure an Apache Kafka topic.
 - a. From the command prompt execute the following command to create a topic named orders.

```
kafka-topics.bat --bootstrap-server localhost:9092 --create --topic  
orders --partitions 2 --replication-factor 1
```

You get a message that the topic is created.

- b. Create a consumer group.

```
kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic  
orders --group ordersconsumer
```

- c. To view the topic created.

```
kafka-topics.bat --zookeeper 127.0.0.1:2181 -list
```

For more details, see Prerequisites for Creating a Connection in *Using the Apache Kafka Adapter with Oracle Integration 3*.

Configure Oracle Autonomous Data Warehouse

Complete configuration tasks in your Oracle Autonomous Data Warehouse instance to successfully connect to it using Oracle Integration and import orders.

The following steps give an overview of the tasks you need to perform to obtain information and details for configuring the Oracle Autonomous Data Warehouse connection from Oracle Integration. For more information on the steps, see *Prerequisites for Creating a Connection in Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3*.

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.

2. 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.

3. 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.

4. Get the database service username and password to connect to the ADW database.

The database service username is the schema username for the user to log in to the database. Note that the database service username is *not* the same as the database service name that you specify in the connection properties on the Connections window.

Additionally, create a database table to write the imported orders. Download the following SQL script - [SQLScript.txt](#), and run it on your Oracle Autonomous Data Warehouse instance to create the required database table.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .

Field	Information to Enter
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Apache Kafka Connection

1. In the Connections section, click the connection name.
2. In the **Bootstrap Servers** field, enter the Bootstrap Server URL you obtained earlier while configuring Apache Kafka. See [Configure Apache Kafka](#).
3. In the **Security Policy** field, select **No Security Policy**.
4. In the Agent Group section, select the agent group.
 - 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 [Configure Apache Kafka](#).
 - c. Click **Use**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Configure the Oracle ADW Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the service name of your Oracle ADW instance. See [Configure Oracle Autonomous Data Warehouse](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Click Upload to upload the credential wallet file that you obtained earlier. See Configure Oracle Autonomous Data Warehouse .
Wallet Password	Enter the password for your credential wallet file.
Database Service Username	Enter your Oracle Autonomous Data Warehouse account username.
Database Service Password	Enter your Oracle Autonomous Data Warehouse account password.

4. Click **Save**. If prompted, click **Save** again.

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle ERP Cloud instance with the Procurement Manager role.
 - b. Create an order.

Click the **Task** drop-down menu, and select **Create Order**.

On successfully creating an order in Oracle ERP Cloud, the first integration flow **Oracle ERP Kafka Order Sync Producer** gets triggered and it receives the order details and writes the same into the Apache Kafka topic. Subsequently, the second integration flow **Oracle Kafka ADW Order Sync Consumer** gets triggered which consumes the order details from the Apache Kafka topic and writes it to the ADW database table.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify that the order details are written to the ADW database table.

Execute the following command:

```
SELECT * FROM ERPUSER.SALES_ORDER_LINEITEMS

SELECT * FROM ERPUSER.SALES_ORDER_HEADER
```

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Apache Kafka Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3*

Import Orders from Shopify to Oracle ERP Cloud

Use this recipe to import orders from Shopify to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a purchase order in Oracle ERP Cloud whenever an order is created in Shopify. In addition, whenever the status of a purchase order in Oracle ERP Cloud is updated as *Closed for receiving*, the recipe updates the corresponding Shopify order's status to *Fulfilled*.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When an order is created in Shopify, the integration flow of the recipe is triggered, and it creates a corresponding purchase order in the Oracle ERP Cloud instance. Basic data such as quantity, price, billing location, shipping location, email, and currency are synchronized between the Shopify and Oracle ERP Cloud platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Shopify
- An account on Shopify with the Administrator role
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Shopify and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration and import orders from Shopify to Oracle ERP Cloud.

Configure Shopify

To configure Shopify, see Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud. Also, create descriptive flexfields to store Shopify Order ID, Shopify Product ID and Shopify Product Variant ID.

Note

To run this recipe, you must have set up tax regimes on your Oracle ERP Cloud instance according to your organization's requirements. See [Tax Regimes](#).

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.

2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user
3. Create a descriptive flexfield to store Shopify Order ID.
 - a. Log in to Oracle ERP Cloud instance.
 - b. In the top-right corner, click the profile image, then select **Setup and Maintenance**.
 - c. In the top-right corner, click the dropdown next to **Setup: Compensation Management**, then select **Procurement**.
 - d. Under Functional Areas, click **Purchasing Foundation**.
 - e. Under Purchasing Foundation, select the task **Manage Purchasing Descriptive Flexfields**.
 - f. Create a global segment. See [Create Global Segments](#).

On the Manage Descriptive Flexfields page, search for the flexfield with Flexfield Code **PO_HEADERS**.
 - g. Under Create Segment, enter the Name of the flexfield as **Shopify Order ID**.
 - h. Under Column Assignment, select the Data Type as **Character**.
 - i. Under Validation, specify the value set. For example: 30 Characters.
 - j. Under Display Properties,
 - i. Select the display type as **Text Box**.
 - ii. Specify the Display Size and Display Height. For example, 10.
 - k. To enable BI reporting, under Business Intelligence, check the **BI Enabled** checkbox.
4. Deploy the descriptive flexfield. See [Deploy Descriptive Flexfields](#).
5. Repeat steps 3 and 4 to create descriptive flexfields that store Shopify Product ID and Shopify Product Variant ID.
 - a. In step 3c, click the dropdown next to **Setup: Product Management**.
 - b. In step 3d, under Functional Areas, click **Items**, then select **Manage Item Descriptive Flexfields**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Shopify Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the Shopify host name. The Shopify host name appears in the Shopify home URL. For example: <code>https://hostname.myshopify.com</code> .
Shopify REST API Version	Enter the REST API version.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Shopify Security Policy selected.
Username	Enter the API key obtained earlier. See Configure Shopify .
Password	Enter the password obtained earlier.
Shared secret	Enter the shared secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Update Integration Property

Update the integration property with appropriate value.

Note

This is an app-driven integration, and you can update the property values only before activating the recipe.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the **EmailTo** integration property with appropriate values.
 - **EmailTo:** This integration property holds the email address to which error notifications are sent. Enter an email address of your choice.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Configure the Lookup Tables

Edit the following lookup tables to map the fields between Shopify and Oracle ERP Cloud.

- **ORACLE-BRT-SHOPIFY_ERP_PO_LOOKUP:** Used to map the fields related to the purchase order between Shopify and Oracle ERP Cloud.
 - **ORACLE-BRT-SHOPIFY_ERP_PO_LINES_LOOKUP:** Used to map the fields related to each line item in the purchase order between Shopify and Oracle ERP Cloud.
 - **ORACLE-BRT-SHOPIFY_ERP_CURRENCY_LOOKUP:** Used to map the currency fields between Shopify and Oracle ERP Cloud.
1. In the Lookups section, click the lookup name.
 2. In the **Property** column, enter the name of the Oracle ERP Cloud field you want to map.
 3. In the **Value** column, enter the name of the field in Shopify to which the Oracle ERP Cloud field has to be mapped.
 4. Click **Save**. If prompted, click **Save** again.
 5. To return to the project workspace, click **Go back** .

Configure the Integration Flow (Optional)

If you want to add or edit the Oracle ERP Cloud fields (other than what is pre-configured in this recipe), you must update all the Stage File elements of the **Oracle Shopify ERP Order2PurchaseOrder Create** integration flow with a new schema file that contains the structure of the updated data set.

Update the Stage File Element

1. Add the required additional fields to your sample schema file.
2. In the Integrations section of the project workspace, click **Actions** ... on the **Shopify ERP Order2PurchaseOrder Create** integration flow, then click **Edit**.

3. On the integration canvas, edit all the Stage File elements and upload the modified file to them, and complete the required mappings. See *Configure a Stage File Action in Using Integrations in Oracle Integration 3*.

Activate and Run the Recipe

After you've configured the connections and other resources, activate the recipe package and run it.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe to create a purchase order in Oracle ERP Cloud. To do so, create a purchase order in Shopify.
 - a. Log in to the Shopify instance.
 - b. On the left navigation pane, click **Orders**.
 - c. To create a new order, click **Create Order**.
 - d. Enter the following details:
 - In the **Customer** field, select the customer for whom the order has to be created.
 - In the **Product** field, select the required product.
 - Modify the quantity as required, and click **Collect Payment**. A pop-up window appears.
 - e. Click **Create Order**.

You've now created an order record in Shopify.

When you create a purchase order in Shopify, the **Oracle Shopify ERP Order2PurchaseOrder Create integration** gets triggered and a draft purchase order gets created in Oracle ERP Cloud followed by an approval process. Once approved, the status of the purchase order in Oracle ERP Cloud changes to *Open*.

3. Run the recipe to update the purchase order status in Shopify. For a purchase order fulfilled in Oracle ERP Cloud, create a receipt.
 - a. Log in to Oracle ERP Cloud instance.
 - b. Click **Navigator > Procurement > My Receipts**.
This takes you to the Receive Items page where you can search for your purchase order.
 - c. On the Receive Items page, in the Purchase Order field, enter the ID of the Purchase Order for which you want to create a receipt.
 - d. Click **Search**.
 - e. In the Search Results section, select the line containing your purchase order.
 - f. Click **Receive**.
This takes you to the Create Receipts page where you can enter the number of items received, waybill, and packing slip number to finalize the receipt.
 - g. On the Create Receipts page, enter the required details.

Note

You can use the **Show Receipt Quantity button** to optionally populate the Quantity field.

- Click **Submit**. Once you click the Submit button, you've created your receipt and you'll receive a confirmation message. The receipt number displays on the confirmation message.
- Click **OK** to close the confirmation window.
Once you create a receipt, the order status now changes to *Closed for receiving*. This action triggers the **Oracle ERP Shopify PO2Order Fulfillment Update** integration flow of the recipe, which updates the corresponding order status in Shopify to *Fulfilled*.
- Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
- Log in to your Oracle ERP Cloud instance and check for the new orders created.
 - On the home page, click **Procurement**.
 - Under Procurement, click **Purchase Orders**.

You will see the recent orders created.
- Log in to your Shopify instance and verify if the status of the fulfilled purchase order is changed to *Fulfilled*.

On the Shopify dashboard, click **Apps**, and then select **Stock Inventory Management**. Check the status of the purchase order.

Related Documentation

- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Import Orders from Shopify to Salesforce

Use this recipe to import orders from Shopify to Salesforce.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates an order in Salesforce whenever an order is created in Shopify.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When an order is created in Shopify, the integration flow of the recipe is triggered, and it creates a corresponding order in Salesforce.

Basic data such as billing city, postal code, province, billing street, and so on are synchronized between the Shopify and Salesforce platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher

- Shopify
- An account on Shopify with the Administrator role
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Shopify and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and transfer orders between them.

Configure Shopify

To configure Shopify, see Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Configure Salesforce

To access Salesforce from Oracle Integration and create orders, you must perform certain configurations 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 order and account records.

Log in to your Salesforce instance as an Administrator and complete the following tasks.

① Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Orders** 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 username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note it down.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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. Create a custom field in Salesforce to store the Shopify Order ID.

This recipe uses unique IDs associated with Shopify and Salesforce instances to synchronize orders and customer records. Create a custom field in Salesforce to store Shopify Order ID.

 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Orders**, and then click **Fields**.
 - b. On the Order Fields page, scroll to the Order Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:

- i. Find and select the **Text** radio button, and click **Next**.
- ii. Enter the following details for the new custom field:
 - Enter ShopifyOrderId 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 Order Fields page, you can see the new field added under the Order Custom Fields & Relationships section.

6. Similarly, create a custom field in Salesforce to store the Shopify Customer ID.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Accounts**, and then click **Fields**.
 - b. On the Account Fields page, scroll to the Account Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter ShopifyCustomer_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 Account Fields page, you can see the new field added under the Account Custom Fields & Relationships section.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Shopify Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the Shopify host name. The Shopify host name appears in the Shopify home URL. For example: <code>https://hostname.myshopify.com</code> .
Shopify REST API Version	Enter the REST API version.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Shopify Security Policy selected.
Username	Enter the API key obtained earlier. See Configure Shopify .
Password	Enter the password obtained earlier.
Shared secret	Enter the shared secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the API key obtained earlier. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Update Integration Property

Update the **EmailTo** integration property with an appropriate value.

Note

This is an app-driven integration, and you can update the property values only before activating the recipe.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the **EmailTo** integration property with appropriate value.
 - **EmailTo:** This integration property holds the email address to which error notifications are sent. Enter an email address of your choice.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Configure the Lookup Table

Edit the lookup table to map the price book names.

1. In the Lookups section, click the lookup name.
2. In the **Property** column, enter the name of the property. For example, `Salesforce_PriceBook_Name`.
3. In the **Value** column, enter the name of the price book to which the property name has to be auto mapped.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
To run the recipe, you must create an order in Shopify.

Note

Ensure that all the products in Shopify are available in Salesforce and are listed under a price book.

- a. Log in to the Shopify instance.
- b. On the left navigation pane, click **Orders**.
- c. To create a new order, click **Create Order**.
- d. Enter the following details:
 - In the **Customer** field, select the customer for whom the order has to be created.
 - In the **Product** field, select the required product.
 - Modify the quantity as required, and click **Collect Payment**. A pop-up window appears.
- e. Click **Create Order**.

You've now created an order record in Shopify and triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Salesforce instance and check for the new orders created.
 - a. On the Salesforce instance, click the **Orders** tab.
 - b. On the Orders Home page, select **All Activated Orders** in the **View** field. Click **Go** if necessary. The list of orders is displayed.
 - c. To view an order record, click its name.

Related Documentation

- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the Salesforce Adapter with Oracle Integration 3*

Import Orders from Zuora to a Hasura Supported Database

Use this recipe to import orders from Zuora to a database supported by Hasura.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports details of orders created in Zuora to a Hasura supported database.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When an order is created in Zuora, the integration flow of the recipe is triggered. The recipe imports the newly created order details from Zuora and subsequently records it in a database supported by Hasura. Basic information such as order category,

account id, account name, order date, order description, order id, order number, and order status are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Zuora
- An account on Zuora with the Administrator role
- Hasura
- An account on Hasura with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Zuora and Hasura instances to connect to them using Oracle Integration, and import order details.

Configure Zuora

To access Zuora using Oracle Integration you must perform some configurations on your Zuora Instance.

1. Log in to your Zuora application with administrator credentials.
2. Create an OAuth client for a user and obtain the client ID and client secret. See [Create an OAuth Client for a User](#).
3. Create an OAuth provider in the Zuora application. See [Add an OAuth 2.0 Provider](#).
4. Pass the Zuora minor version (for example 224.0) in the request header in the operations which require it.
5. Create order notification.
 - a. Click the arrow adjacent to your profile name.
 - b. In the drop-down list that opens, select **Billing**.
 - c. Under the Billing Settings section, Select **Setup Profiles, Notifications and Email Templates**.
 - d. Click **Add New Notification**.

The Add notification window appears.
 - e. In the **Related Event** drop-down list, select **Order Processed**.
 - f. Enter the **Name** and **Description** for the Notification and select the **Active** checkbox.
 - g. Under the **Delivery Options**:
 - i. Select the **Email** and **Callout** checkboxes.
 - ii. Update the **Base URL** once you activate the recipe.
 - iii. Leave **HTTP Method** as **POST**.
 - iv. Add the parameter details. as shown in the table below.

Parameter name	Parameter value
AccountAccountNumber	Account.AccountNumber

Parameter name	Parameter value
AccountId	Account.Id
AccountName	Account.Name
OrderCategory	Order.Category
OrderDescription	Order.Description
OrderId	Order.Id
OrderOrderDate	Order.OrderDate
OrderOrderNumber	Order.OrderNumber
OrderStatus	Order.Status

- v. Select your OAUTH 2.0 provider.
- h. Click **Add New Notification**.

Configure Hasura

To access Hasura using Oracle Integration and import orders from Zuora, you must perform certain configuration tasks on your Hasura instance.

Log in to your Hasura instance as an **Administrator** and perform the following tasks.

1. Create a new project.
 - a. Click **New Project**.
 - b. Hasura auto-creates few basic information for this project. For example: Name, ID, GraphQL API, Admin Secret and so on.
 - c. You can fill the required details in General, Env vars, Domains, Git Deploy, Usage, Integrations, and Alerting sections.

Amongst all the configuration details entered above, GraphQL API and Admin Secret will be required to establish a connection with the GraphQL Adapter.
2. Connect to a new or existing database.
 - a. Click **Launch Console** to open the Hasura console for the project.
 - b. Navigate to **Data**, then **Manage**, and then click **Connect Database**.
 - c. Select an existing database from the list or create a new database (for example: Postgres).
 - d. Create a table with the columns: account_id, account_name, order_id, order_category, order_number, order_date, order_description and order_status.

For more details, see [Quickstart with Hasura Cloud](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Zuora Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the host name of the Zuora instance. The Zuora host name appears in the Zuora home URL. For example: `https://hostname.zuora.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Zuora Composite Security Policy .
Client Id	Enter the client ID retrieved from the Zuora instance. See Configure Zuora .
Client Secret	Enter the Client Secret retrieved from Zuora instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle GraphQL Hasura Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the GraphQL endpoint which you have retrieved from Hasura console. See [Configure Hasura](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Custom Security Policy .
Header Key	Enter X-Hasura-Admin-Secret. See Admin Access .
Header Value	Enter the admin secret of the project created on Hasura as the header value. See Configure Hasura .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Update Integration Properties

Update integration properties for the integration flow of the recipe with appropriate values.

1. In the Integrations section, click **Actions ...** on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **EmailTo**: Specify the email ID to which the error notification has to be sent.
 - b. **retryCount**: Specify the number of retries when an invoke fails. The default value is 3.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Get the integrations's endpoint URL.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run details**.
 - b. In the panel that opens, copy the Metadata URL. This is the integration's endpoint URL.
3. Update the integration's endpoint URL in your Zuora instance.
 - a. Log in to your Zuora instance.
 - b. Navigate to **Billing**, select **Setup Profiles**, and then select **Notifications and Email Templates**.
 - c. From the list that appears, edit the notification which you had created against this order. See [Configure Zuora](#).
 - d. In the Edit Notification window that opens, enter the endpoint URL that you obtained in step 2 as the Base URL.

Note

The endpoint URL changes with respect to the Oracle Integration instance.

- e. Click **Save**.
4. Run the recipe.

To run the recipe, you must create an order in the Zuora instance.

 - a. Under **Customers**, select **Orders**.
 - b. In the **Orders** tab that appears, either click **Create Subscription** or **Amend Subscription**.
 - c. Enter the product, its rate plan, charges and then click **Add Product**.
 - d. Click **Review Order**.

Note

You can only review your order when you are creating a subscription. This option is not available when you are amending an existing subscription.

- e. Click **Activate**.

You've successfully created an order and triggered the integration flow of the recipe.
5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Check if the details of the new order created in Zuora have been imported into the Hasura supported database.

- a. Log in to your Hasura instance.
- b. Navigate to **API Explorer**.
- c. Fetch the order details from the database by passing the order id or order number.

Related Documentation

- *Using the GraphQL Adapter with Oracle Integration 3*
- *Using the Zuora Adapter with Oracle Integration 3*

Import Products from Salesforce to Shopify

Use this recipe to import products from Salesforce to Shopify.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a corresponding product in Shopify whenever an item is created in Salesforce.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a product create event is generated in Salesforce, the integration flow of the recipe is triggered, and it creates a corresponding product in the Shopify store using the GraphQL adapter. Basic data such as vendor, product type, and status are synchronized between the Salesforce and Shopify platforms.

System and Access Requirements

- Oracle Integration, Version 24.08 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Shopify
- An account on Shopify with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Shopify instances to successfully connect to these external systems using Oracle Integration and import products from Salesforce to Shopify.

Configure Salesforce

Log in to your Salesforce instance as an **Administrator** and perform the following configuration tasks.

Obtain the Salesforce Domain URL, API Version, Client ID and Client Secret. See *Prerequisites for Creating a Connection in Using the Salesforce REST Adapter with Oracle Integration 3*.

In addition, you must create a custom field for Shopify Product ID in Salesforce to hold the Shopify ID. This recipe uses unique Shopify Product ID associated with product synced in Shopify from Salesforce.

Create a custom field for Shopify Product ID:

1. On the Salesforce Setup page:
 - a. Scroll to the Build section in the left navigation pane.
 - b. Expand **Customize**, then **Products**, and then click **Fields**.
2. On the Product Fields page, scroll to the Product Custom Fields & Relationships section and click **New**.
3. On the New Custom Field page:
 - a. Find and select the **Text** radio button and click **Next**.
 - b. Enter the following details for the new custom field, and then click **Next**.
 - i. Enter `Shopify_Product_Id` as the field label.
 - ii. Enter `Shopify` as the field name.
 - iii. Enter `200` as the length.
 - iv. Optionally, enter a description for the new field.
 - v. Select the **External ID** check box.
 - c. 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**.
 - d. Leave all the layout check boxes selected and click **Save**.

On the Product Fields page, you can see the new field added under the Product Custom Fields & Relationships section.

Configure Shopify

To configure Shopify, see *Prerequisites for Creating a Connection in Using the Shopify Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Salesforce REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Salesforce Domain URL	Enter the Salesforce Domain URL. See Configure Salesforce .
API Version	Select 60 .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Client Credentials .
Client ID	Enter the client ID of the account created for Oracle Integration on the Salesforce instance. See Configure Salesforce .
Client Secret	Enter the client secret of the account created for Oracle Integration on the Salesforce instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the GraphQL Shopify Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the GraphQL Endpoint URL. For example: `https://{shop}.myshopify.com/admin/api/2024-04/graphql.json`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Custom Security Policy .
Header Key	Select the X-Shopify-Access-Token .
Header Value	Enter the Header Value (API access token). See Configure Shopify .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORACLE-BRT-SHOPIFY_PROPERTIES** lookup table and enter appropriate values.

1. In the Lookups section, click the lookup name.
2. In the lookup table, map the lookup key with an appropriate value as shown below.

Key	Description	Value
status	Specifies the status of the product created on Salesforce.	Active

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Update Integration Properties

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **EmailTo**: This integration property holds the email address to which error notifications are to be sent.
 - b. **retryCount**: This integration property holds the number of retries when an invoke fails. The default value is 3.

3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you have to create a product in Salesforce.

- a. Log in to your Salesforce instance.
- b. On the Salesforce Setup page, click the **Products** tab.
- c. On the **Products** Home page, click **New**.
- d. On the New Product page, enter the product details, such as Product Name, Product Code, Status (Active), Product Family, and Product Description (Optional). Click **Save**.

You've now created a product in Salesforce and successfully triggered the integration flow of the recipe for execution.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Shopify instance and check for the new product created.
 - a. On the home page, select your Shopify store, for example, `orcl-shop-connector-dev-store`.

- b. On the store page, click **Products** on the left navigation pane.

The list of products is displayed, which contains the corresponding product record for the item created in Salesforce.

Related Documentation

- *Using the Salesforce REST Adapter with Oracle Integration 3*
- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the GraphQL Adapter with Oracle Integration 3*

Import Purchase Orders from Oracle ERP Cloud to Microsoft SQL Server

Use this recipe to import purchase order details (and associated customer data) from Oracle ERP Cloud to a Microsoft SQL Server database.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that saves the order details (and associated customer data) in a Microsoft SQL Server database whenever a purchase order is created in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a purchase order is created in Oracle ERP Cloud, the integration flow of the recipe is triggered, and it saves the purchase order details in a Microsoft SQL Server database.

System and Access Requirements

- Oracle Integration, Version 23.8.0.0.0 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- A Microsoft SQL Server database
- An account on the Microsoft SQL Server database with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Microsoft SQL Server instances in order to successfully connect to these external systems using Oracle Integration and import purchase order details.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure the Microsoft SQL Server Database

To access the Microsoft SQL Server database from Oracle Integration and import customer and purchase order details, you must perform certain configurations on your Microsoft SQL Server instance. See *Prerequisites for Creating a Connection in Using the Microsoft SQL Server Adapter with Oracle Integration 3*.

In addition, log in to your Microsoft SQL Server Management Studio and add tables according to the following information. See [Create Tables \(Database Engine\)](#).

1. Create Customer table with the following columns.

Column	Datatype
BuyerId	varchar(255)
 Note Select this variable as the Primary Key (PK) and Not Null (NN).	
PersonName	varchar(255)
Gender	varchar(255)
Email	varchar(255)
WorkPhone	varchar(255)
Address1	varchar(255)
Address2	varchar(255)
Building	varchar(255)
City	varchar(255)
Province	varchar(255)
PostalCode	varchar(255)
Country	varchar(255)

2. Similarly, create another table named PurchaseOrder, and add the following columns.

Column	Datatype
POHeaderId	varchar(255)

 **Note**
 Select this variable as the **Primary Key (PK)** and **Not Null (NN)**.

OrderNumber	varchar(255)
ProcurementBU	varchar(255)
BuyerId	varchar(255)
Buyer	varchar(255)
Ordered	varchar(255)
Supplier	varchar(255)
SupplierId	varchar(255)
Currency	varchar(255)
DiscountAmount	varchar(255)
TotalTax	varchar(255)
Total	varchar(255)

3. Create another table named PurchaseOrderDetail, and add the following columns.

Column	Datatype
POHeaderId	varchar(255)

 **Note**
 Select this variable as the **Primary Key (PK)** and **Not Null (NN)**.

OrderNumber	varchar(255)
POLineId	varchar(255)
LineNumber	varchar(255)
LineType	varchar(255)
Quantity	varchar(255)
Item	varchar(255)
SupplierId	varchar(255)
ItemId	varchar(255)
ItemDescription	varchar(255)
Price	varchar(255)

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on the Oracle ERP Cloud instance. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle ERP Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Microsoft SQL Server Database

To access the Microsoft SQL Server database from Oracle Integration and import customer and purchase order details, you must perform certain configurations on your Microsoft SQL Server instance. See *Prerequisites for Creating a Connection in Using the Microsoft SQL Server Adapter with Oracle Integration 3*.

In addition, log in to your Microsoft SQL Server Management Studio and add tables according to the following information. See [Create Tables \(Database Engine\)](#).

1. Create Customer table with the following columns.

Column	Datatype
BuyerId	varchar(255)

Note

Select this variable as the **Primary Key (PK)** and **Not Null (NN)**.

PersonName	varchar(255)
Gender	varchar(255)
Email	varchar(255)
WorkPhone	varchar(255)
Address1	varchar(255)
Address2	varchar(255)

Column	Datatype
Building	varchar(255)
City	varchar(255)
Province	varchar(255)
PostalCode	varchar(255)
Country	varchar(255)

2. Similarly, create another table named PurchaseOrder, and add the following columns.

Column	Datatype
POHeaderId	varchar(255)

Note

Select this variable as the **Primary Key (PK)** and **Not Null (NN)**.

OrderNumber	varchar(255)
ProcurementBU	varchar(255)
BuyerId	varchar(255)
Buyer	varchar(255)
Ordered	varchar(255)
Supplier	varchar(255)
SupplierId	varchar(255)
Currency	varchar(255)
DiscountAmount	varchar(255)
TotalTax	varchar(255)
Total	varchar(255)

3. Create another table named PurchaseOrderDetail, and add the following columns.

Column	Datatype
POHeaderId	varchar(255)

Note

Select this variable as the **Primary Key (PK)** and **Not Null (NN)**.

OrderNumber	varchar(255)
POLineId	varchar(255)
LineNumber	varchar(255)
LineType	varchar(255)
Quantity	varchar(255)
Item	varchar(255)
SupplierId	varchar(255)
ItemId	varchar(255)

Column	Datatype
ItemDescription	varchar(255)
Price	varchar(255)

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
Create a purchase order in Oracle ERP Cloud.
 - a. Log in to your Oracle ERP Cloud instance with the Sales User role.
 - b. Click **Procurement**, then **Purchase Orders**, and then **Tasks**.
 - c. Under **Orders**, click **Create Order**.
 - d. In the Create Order window, specify the details of the purchase order and click **Create**.
 - e. Further, add line items to the purchase order and click **Submit**.You've now successfully triggered the recipe.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to the Microsoft SQL Server Management Studio and verify if the purchase order details and associated customer data are populated in the database tables.

In SQL Server Object Explorer, right-click any of the tables you created earlier and select **View Data**.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Microsoft SQL Server Adapter with Oracle Integration 3*

Import Purchase Orders from Oracle JD Edwards EnterpriseOne to Oracle ERP Cloud

Use this recipe to import purchase orders from Oracle JD Edwards EnterpriseOne to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a purchase order in Oracle ERP Cloud whenever an order is created in Oracle JD Edwards EnterpriseOne.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When an order is created in Oracle JD Edwards EnterpriseOne, the integration flow of the recipe is triggered, and it creates a corresponding purchase order in the Oracle ERP Cloud instance. Basic data such as quantity, price, billing location, shipping location, email, and currency are synchronized between the Oracle JD Edwards EnterpriseOne and Oracle ERP Cloud platforms.

System and Access Requirements

- Oracle Integration, Version 24.06 or higher
- Oracle JD Edwards EnterpriseOne
- An account on Oracle JD Edwards EnterpriseOne with the Administrator role
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle JD Edwards EnterpriseOne and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration import purchase orders from Oracle JD Edwards EnterpriseOne to Oracle ERP Cloud.

Configure Oracle JD Edwards EnterpriseOne

To configure Oracle JD Edwards EnterpriseOne, see *Prerequisites for Creating a Connection in Using the Oracle JD Edwards EnterpriseOne Adapter with Oracle Integration 3*. Also, import the zip file into the Orchestrator Studio and configure the REST connection.

1. Import the zip file into the Orchestrator Studio.
 - a. Sign in to the Orchestrator Studio. (You can access EnterpriseOne from the Tools page in the Studio.)
 - b. Click the **Tools** link in the upper right corner.
 - c. Select **Import/Export**, and import the following zip file: [LaunchOICG3Flow.zip](#).
2. Configure the REST connection.
 - a. In Orchestrator studio, click the **Tools** link.
 - b. On the Tools page, click the **Connections** icon.
 - c. Click the REST connection and enter details for the following fields in the Connection Information area:
 - **Name:** Enter the name of the REST Connection. For example, OICRESTENDPOINT.
 - **Description:** Enter a short description of the connection.
 - **Type:** Select REST.
 - **User/Role:** Enter *PUBLIC.
 - **Environment:** Enter the environment of your Oracle JD Edwards EnterpriseOne instance.
 - d. On the Service Information tab, enter the OIC REST trigger URL in the Endpoint field.
 - e. On the Security tab, enter the following details.

Field	Information to Enter
Security Policy	Select a security policy based on your Oracle Integration orchestration REST adapter configuration. For example, Basic Authorization .
Username	Enter the username of your Oracle Integration account.
Password	Enter the password of your Oracle Integration account.

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.

Note

To run this recipe, you must have set up tax regimes on your Oracle ERP Cloud instance according to your organization's requirements. See [Tax Regimes](#) in *Implementing Tax*.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle JD Edwards EnterpriseOne Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
WSIL URL	Select Enter the WSIL URL. For example, <code>https://Business_Services_Server:port/Environment/wsil.jsp</code> See Configure Oracle JD Edwards EnterpriseOne .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave JD Edwards Security Policy selected.
Username	Enter the username of your Oracle JD Edwards EnterpriseOne account. See Configure Oracle JD Edwards EnterpriseOne .
Password	Enter the password of your Oracle JD Edwards EnterpriseOne account.
Environment	Enter the environment obtained earlier.
Role	Enter the role type that can access the JD Edwards EnterpriseOne environment. To select all the roles, enter *ALL.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

Edit the following lookup tables to map the fields between Oracle JD Edwards EnterpriseOne and Oracle ERP Cloud instances.

- **ORACLE-BRT-SHOPIFY_ERP_PO_LOOKUP**: Used to map the fields related to the purchase order between Oracle JD Edwards EnterpriseOne and Oracle ERP Cloud.

- **ORACLE-BRT-SHOIFY_ERP_PO_LINES_LOOKUP**: Used to map the fields related to each line item in the purchase order between Oracle JD Edwards EnterpriseOne and Oracle ERP Cloud.
 - **ORACLE-BRT-SHOIFY_ERP_SHIPTO_LOOKUP**: Used to map the ShipToLocation and ShipToOrganization fields in Oracle ERP Cloud with ShipTo field in Oracle JD Edwards EnterpriseOne.
 - **ORACLE-BRT-SHOIFY_ERP_SUPPLIER_LOOKUP**: Used to map the Supplier and SupplierSite fields in Oracle ERP Cloud with SupplierEntityId field in Oracle JD Edwards EnterpriseOne.
 - **ORACLE-BRT-SHOIFY_ERP_CATEGORY_LOOKUP**: Used to map the Product Category in Oracle ERP Cloud with Product Catalog field in Oracle JD Edwards EnterpriseOne.
 - **ORACLE-BRT-SHOIFY_ERP_BUYER_LOOKUP**: Used to map the Buyer field in Oracle ERP Cloud with BuyerEntityId field in Oracle JD Edwards EnterpriseOne.
 - **ORACLE-BRT-SHOIFY_ERP_UOM_LOOKUP**: Used to map the UOM field in Oracle ERP Cloud with EOM field in Oracle JD Edwards EnterpriseOne.
1. In the Lookups section, click the lookup name.
 2. In the **Name** column, enter the Oracle JD Edwards EnterpriseOne field that you want to map.
 3. In the **Value** column, enter the Oracle ERP Cloud field to which the Oracle JD Edwards EnterpriseOne field must be mapped.
 4. Click **Save**. If prompted, click **Save** again.
 5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe. Create a purchase order in Oracle JD Edwards EnterpriseOne.
 - a. Sign in to your Oracle JD Edwards EnterpriseOne instance.
 - b. On the top navigation pane, click **EnterpriseOne Menu**, and select **Supply Management**. Further, choose **Procurement and Subcontract Management**, and select **Daily Processing**.
 - c. Click **Stock Based Purchasing** and select **Purchase Order Processing** and then click **Enter Purchase Orders**.
 - d. To create a new purchase order, click **Add**.
 - e. Enter the values for the following fields: **Branch/Plant**, **Supplier**, **Ship To**, **Buyer and Carrier**.
 - f. Click **OK**.
 - g. Enter the line items. For each line item, enter the **Item Number**, **Quantity**, and **Cost**.

You've now created a purchase order record in Oracle JD Edwards EnterpriseOne. When you create a purchase order in Oracle JD Edwards EnterpriseOne, the integration flow gets triggered, and a purchase order gets created in Oracle ERP Cloud. The status of the purchase order in Oracle ERP Cloud changes to either Pending Approval or Open.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle ERP Cloud instance and check for the new orders created.
 - a. On the home page, click **Procurement**.
 - b. Under Procurement, click **Purchase Orders**.

You will see the recent orders created.

Related Documentation

- *Using the Oracle JD Edwards EnterpriseOne Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Import Sales Orders from Oracle ERP Cloud to MySQL

Use this recipe to import sales order details from Oracle ERP Cloud to a MySQL database.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that saves the order details in a MySQL database whenever a sales order is created in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a sales order is created in Oracle ERP Cloud, the integration flow of the recipe is triggered, and it saves the sales order details in a MySQL database.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- A MySQL database
- An account on the MySQL database with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and MySQL instances in order to successfully connect to these external systems using Oracle Integration and import sales order details.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure the MySQL Database

To access the MySQL database from Oracle Integration and import order details, you must perform certain configurations on your MySQL instance. See *Prerequisites for Creating a Connection in Using the MySQL Adapter with Oracle Integration 3*.

In addition, create a database schema and add tables according to the following information:

1. Log in to your MySQL Workbench instance as an **Administrator**.
2. Click **File** and select **Create Schema**.
3. Specify the name of the schema as `oic_schema` and click **Apply**.
4. In the **Apply SQL Script to Database** window, click **Apply** to run the SQL command.
5. Click **Finish**. You can view the new schema with no tables on the left pane.
6. Expand `oic_schema` on the left pane. Right-click **Tables**, and select **Create Table....**
7. Specify the name of the table as `erp_salesorder`.
8. Double-click the empty table to add the following columns.

Column	Datatype
ERP_ORDER_NUMBER	VARCHAR(40)

Note

Select this variable as the **Primary Key (PK)** and **Not Null (NN)**.

CreationDate	DATETIME
Contact_FirstName	VARCHAR(40)
Contact_LastName	VARCHAR(40)
OrderTotal	VARCHAR(40)
CustomerName	VARCHAR(40)
CustomerId	VARCHAR(40)
CurrencyCode	VARCHAR(40)

9. Similarly, create another table named `erp_salesorder_lines`, and add the following columns.

Column	Datatype
ERP_ORDER_NUMBER	VARCHAR(40)
Note Select this variable as the Primary Key (PK) and Not Null (NN).	
ProductQty	VARCHAR(45)
Note Select this variable as the Primary Key (PK) and Not Null (NN).	
ProductName	VARCHAR(45)
Note Select this variable as the Primary Key (PK) and Not Null (NN).	
ProductDescription	VARCHAR(100)
Note Select this variable as the Primary Key (PK) and Not Null (NN).	

10. Click **Apply**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on the Oracle ERP Cloud instance. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle ERP Cloud instance.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the MySQL Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the MySQL server host name.
Port	Enter the MySQL port.
Database Name	Enter <code>oic_schema?sslMode=DISABLED</code> .

- In the Security section, enter the following details:

Field	Information to Enter
Username	Enter the username of your MySQL instance.
Password	Enter the password of your MySQL instance.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Update Integration Properties

Note

As the recipe contains an app-driven integration, you can update the integration properties only before activating the recipe.

- In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.

2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **ToEmail:** This integration property holds the email address to which error notifications are sent. Enter an email address of your choice.
 - b. **FromEmail:** This integration property holds the email address from which error notifications are sent. Enter an email address of your choice.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

Create a sales order in Oracle ERP Cloud. See [Create Sales Orders](#) in *Using Order Management*.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your MySQL instance and verify if the sales order details are populated in the database table.

Right-click the `erp_salesorder` table from the left pane and select **Select Rows - Limit 1000**.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the MySQL Adapter with Oracle Integration 3*

Import Sales Orders from Oracle NetSuite to Oracle Autonomous Data Warehouse

Use this recipe to import sales orders from Oracle NetSuite to Oracle Autonomous Data Warehouse (ADW).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports sales orders created in Oracle NetSuite and loads them into Oracle Autonomous Data Warehouse database.

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 `sinceDateTime` to query sales orders created on or after the date and time specified in the parameter. The recipe then extracts the sales orders from Oracle NetSuite and writes them into Oracle Autonomous Data Warehouse database tables.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle NetSuite, Version 2023.1.0
- Oracle Autonomous Data Warehouse, Oracle Database 19c Enterprise Edition
- An Autonomous Database instance created in Oracle Cloud Infrastructure (OCI)
- An account in Oracle NetSuite with the Administrator role
- An account in Oracle Autonomous Data Warehouse with the Administrator role
- An Oracle Autonomous Data Warehouse database table with write access

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite instance and Oracle Autonomous Data Warehouse instance in order to successfully connect to these external systems using Oracle Integration and import sales orders.

Configure Oracle NetSuite

To access Oracle NetSuite using Oracle Integration and import sales orders, you must perform certain configuration tasks on your Oracle NetSuite instance.

To successfully connect to Oracle NetSuite, you must perform certain general configurations on your Oracle 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 Oracle NetSuite saved search. The search criterion specified in the saved search will be used to fetch the sales orders that are to be imported to Oracle Autonomous Data Warehouse.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See [Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).
2. Perform the TBA-related configuration tasks. See [Prerequisites for the Token-Based Authentication Security Policy in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Lists	Sales Order	Full
Lists	Subsidiaries	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Create a saved search to get the sales records that satisfy the search criterion. The sales records that are the output of the search result will be imported into Oracle Autonomous Data Warehouse.
 - a. Navigate to the New Saved Search window using any one of the following navigation options:
 - On the Oracle NetSuite home page, select **Reports**, then choose **Saved Searches**, then **All Saved Searches**, and then **New**.
 - On the Oracle NetSuite home page, select **Transactions**, then choose **Management**, then **Saved Searches**, and then **New**.
 - b. On the New Saved Search window, select the object type to perform the search operation. For example, select **Transaction**.
 - c. Add a title and ID for the saved search in the **SEARCH TITLE** and **ID** fields respectively.

The title and ID for the saved search will help you in keeping track of the saved search when you want to reuse it to re-query latest data that matches the search criteria.
 - d. Set the search criteria conditions. Select the **Criteria** tab and specify the conditions. For example, you can specify a condition to filter sales orders created after a certain date.

If the **USE EXPRESSIONS** check box is selected, you can also use additional options such as AND, OR, NOT.
 - e. Configure the search results that are to be displayed. Select the **Results** tab, and in the **Columns** tab of the page, select the fields that you want to be displayed in the search result.

Configure Oracle Autonomous Data Warehouse

To access Oracle Autonomous Data Warehouse using Oracle Integration, you must perform certain configuration tasks on your Oracle Autonomous Data Warehouse instance.

The following steps give an overview of the tasks you need to perform to obtain required information for configuring the Oracle Autonomous Data Warehouse connection properties 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 3*.

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.
2. Get the Oracle Autonomous Data Warehouse 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 Autonomous Data Warehouse connection security from Oracle Integration.
3. Get the Oracle Autonomous Data Warehouse database service name. You need the database service name when configuring the connection properties on the Connections window.

Note that the service name must be same as the one in the `tnsnames.ora` file in the client credential wallet.
4. Get the database service username and password to connect to Oracle Autonomous Data Warehouse database.

The database service username is the schema username for the user to log in to the database. Note that the database service username is *not* the same as the database service name that you specify in the connection properties on the Connections window.

5. For bulk data import operation, you must use the JDBC With OCI Signature security policy on the Connections window. Before you can successfully configure this security policy on the Connections window, you must complete the following prerequisites.
 - Create an object storage bucket in Oracle Cloud Infrastructure. Note the region where your object storage is located.
 - Obtain the OCID value of the compartment in which you create the storage bucket.
 - Obtain the tenancy OCID and user OCID details in the Oracle Cloud Console.
 - Create an API signing key in the Oracle Cloud Console.
 - Note the finger print that was generated when you created the key in the Oracle Cloud Console.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle NetSuite **WSDL URL**. For example, `https://webservices.netsuite.com/wsdl/<NetSuite_application_version>/netsuite.wsdl`, where `<NetSuite_application_version>` is the version of the Oracle NetSuite application (for example, `v2015_1_0`). See [Configure Oracle NetSuite](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier.

Note

You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.

- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle ADW Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the service name of your Oracle Autonomous Data Warehouse instance. See [Configure Oracle Autonomous Data Warehouse](#).
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC with OCI Signature .
Wallet	Click Upload and upload the credential wallet file. See Configure Oracle Autonomous Data Warehouse .
Wallet Password	Enter the password for the credential wallet file.
Database Service Username	Enter the username of your Oracle Autonomous Data Warehouse account.
Database Service Password	Enter the password of your Oracle Autonomous Data Warehouse account.
Object Storage Region	Enter the region in which your object storage is located (for example, us-ashburn-1).
Object Storage Tenancy OCID	Specify the value you copied from the Oracle Cloud Console (for example, <code>ocid1.tenancy.oc1.alphanumeric.value</code>).
Object Storage Compartment OCID	Specify the value you copied from the Oracle Cloud Console (for example, <code>ocid1.compartment.oc1.alphanumeric.value</code>).
Object Storage User OCID	Specify the value you copied from the Oracle Cloud Console (for example, <code>ocid1.user.oc1.alphanumeric.value</code>).
Private Key	Click Upload and upload the key you created in the Oracle Cloud Console.
Finger Print	Enter the finger print that was generated when you created the key in the Oracle Cloud Console.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the recipe.

- a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the `sinceDateTime` parameter. This parameter fetches all the sales orders created on or after the date and time specified and loads them into Oracle Autonomous Data Warehouse. Enter a value, for example, `2022-09-02T00:00:00Z` in the **New Value** field to fetch all sales records from Oracle NetSuite created on or after the date specified. The parameter's value is automatically updated after each successful run.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

 Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to the Oracle Autonomous Data Warehouse instance and check if sales orders from Oracle NetSuite have been imported into the Oracle Autonomous Data Warehouse database table.

Related Documentation

- [Using the Oracle NetSuite Adapter with Oracle Integration 3](#)
- [Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3](#)

Import Segments from Oracle Unity to Salesforce Marketing Cloud

Use this recipe to import segments from Oracle Unity to data extensions in Salesforce Marketing Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports segments from Oracle Unity to data extensions in Salesforce Marketing Cloud, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow fetches a segment (multiple records) from Oracle Unity and exports it to a specific location in FTP. Subsequently, the recipe reads the segment from the FTP location and imports it to a data extension in Salesforce Marketing Cloud.

Note

This recipe synchronizes any segment from Oracle Unity to Salesforce Marketing Cloud. If you require to specifically synchronize Customer segments from Oracle Unity to Salesforce Marketing Cloud, you can use a different recipe available in the Integration Store. See [Sync Customer and Campaign Data Between Oracle Unity and Salesforce Marketing Cloud](#).

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle Unity
- An account on Oracle Unity with the Administrator role
- Salesforce Marketing Cloud
- An account on Salesforce Marketing Cloud with the Administrator role
- An account on the Salesforce Marketing Cloud FTP server
- An FTP client to access the Salesforce Marketing Cloud FTP server

Assumptions

- Salesforce Marketing Cloud API request quota is sufficient to support the number of create and update requests from the batch integration.
- Salesforce Marketing Cloud has the data extension already created to which the Oracle Unity records must be imported.

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Unity and Salesforce Marketing Cloud instances in order to successfully connect to these external systems using Oracle Integration and import segments.

Configure Oracle Unity

Complete the following tasks in your Oracle Unity instance to successfully connect to it from Oracle Integration and export segments.

1. Obtain the client ID and client secret. See Prerequisites for Creating a Connection in *Using the Oracle Unity Adapter with Oracle Integration 3*.
2. Create segments. See [Creating Segments](#) in Oracle Unity documentation.
3. Add personalization attributes to your segments. See [Managing Personalization Attributes](#) in Oracle Unity documentation.
4. Create a destination to export segments from Oracle Unity. See [Creating destinations](#) in Oracle Unity documentation.
5. Create a campaign and set up an export of a segment to a marketing orchestration platform. See [Creating campaigns](#) in Oracle Unity documentation.

Note

Ensure that the file name in the campaign is the same as the name of the segment.

6. Run the campaign. See [Managing Campaigns](#) in Oracle Unity documentation.

Note

Ensure that the campaign job is complete. See [Managing the Jobs dashboard](#) in Oracle Unity documentation.

Configure Salesforce Marketing Cloud

To access Salesforce Marketing Cloud from Oracle Integration and import segments, you must perform certain configurations on your Salesforce Marketing Cloud instance.

1. Add a user account to the Salesforce Marketing Cloud SFTP Accounts.
2. Create an FTP location for the Salesforce Marketing Cloud files.
3. Obtain the SOAP WSDL endpoint, username, and password.

Log in to your Salesforce Marketing Cloud instance as an **Administrator** and execute the following tasks.

1. Add a user account to the Salesforce Marketing Cloud SFTP Accounts.
 - a. Hover over your name in the app switcher and click **Setup**.
 - b. Click **Data Management**.
 - c. Click **FTP Accounts**.
 - d. Click **Add FTP User**. By default, the username is the MID number for your current Marketing Cloud MID, including the current parent account or child business unit.
 - e. Enter an email address for the new FTP user.
 - f. Enter an initial password for the user. Password complexity requirements combine Marketing Cloud password policy and server-side FTP password requirements. These policies require a minimum of 12 characters and no reuse of the most recent password.
 - g. Reenter the initial password for the user.
 - h. Select **Full** for User Permissions.
 - i. To restrict account access to specific IP addresses, enter each address in the **Whitelist Ips** field and click **Add**.

This field accepts matching, wildcard, range, and mask values. To allow all IP addresses, enter *. *.*.*.*
 - j. Click **Next**.
 - k. Choose the authentication option for the SFTP account.
 - Password
 - SSH Key
 - SSH Key or Password

- SSH Key and Password
To use any of the SSH key options, ensure that you upload an SSH key in Key Management.
 - l. Choose the SSH key you want to use in the **SSH Keys** field.
 - m. Save your changes. By default, this process creates an enabled, active user as unlocked with a valid current password.
2. Create an FTP location for the Salesforce Marketing Cloud files.
- a. Hover over your name and click **Setup**.
 - b. In the Data Management section, click **File Locations**.
 - c. Click **Create**.
 - d. Complete the information in the Properties section.
 - **Name**: Enter a unique name that's recognizable when creating an activity in Marketing Cloud.
 - **External key**: Enter a unique key that identifies the file location while using the API.
 - **Description**: Enter details that help further identify the file location when creating an activity in Marketing Cloud.
 - e. Select the location type as **Enhanced FTP Site Import Directory** which is a standard directory for file imports.
 - f. Save the file location.
3. Obtain the SOAP WSDL endpoint, username, and password. To do so, you must create a user and assign role and permissions.
- a. Create a user.
 - i. Click **Setup**, select **Users**, and then **Users**.
 - ii. On the Setup Users page, click **Create**.
 - iii. On the New User page:
 - Enter the name of the user.
 - Enter the email ID of the user in **Reply Email Address** field and verify it.
 - Enter the email address to which run-time exception emails are sent in the **Notification Email Address** field.
 - Enter the username and note the same.
 - Select the appropriate **Time Zone** and **Culture Code**.
 - Enable the **API User** option.
 - Note the **SOAP WSDL URL**.
 - Set a password and note the same.
 - b. Assign the required role and permissions to the user.
 - i. On the Setup Users page, select the user you created previously.
 - ii. Click **Manage Roles**.
 - iii. Select whether you assign the role directly to the user or in the context of a specific business unit in the **Business Unit** drop-down menu.
 - iv. Click **Edit Roles**. Select the check box next to the appropriate roles to assign.

- v. Click **Save**.
- vi. Click **Edit Permissions** to update specific permissions for the user and select the check boxes next to permissions to assign.
- vii. Click **Save**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Oracle Unity Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the URL for connecting to Oracle Unity. For example, <code>https://XXXXXX.cxunity.ocs.oraclecloud.com</code> . The URL is provided when you subscribe to Oracle Unity.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Custom Two-Legged Flow . See Configure Connection Security in <i>Using the REST Adapter with Oracle Integration 3</i> .

4. Click **OK**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. A dialog prompts you to select the type of connection testing to perform:
 - **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL.
 - **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.
7. Wait for a message about the results of the connection test.
 - If the test was successful, then the connection is configured properly.
 - If the test failed, then edit the configuration details you entered. Check for typos, verify URLs and credentials, and download the diagnostic logs for additional details. Continue to test until the connection is successful.
8. When complete, click **Save**.

- To return to the Configuration Editor, click **Go back** .

Configure the FTP Salesforce Marketing Cloud Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your SFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

- 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.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the Configuration Editor, click **Go back** .

Configure the SOAP Salesforce Marketing Cloud Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the SOAP WSDL URL in the **WSDL URL** field. See [Configure Salesforce Marketing Cloud](#).
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username. See Configure Salesforce Marketing Cloud .
Password	Enter the password.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. A dialog prompts you to select the type of connection testing to perform:
 - Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL.
 - Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

6. To return to the Configuration Editor, click **Go back** .

Configure the Lookup Tables

1. In the Lookups section, click the lookup name.
2. Enter the field names and corresponding values. Refer to the details provided for each table below.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

The recipe contains the following four lookup tables. Edit them as necessary.

- **ORACLE-BRT-UNITY_SFMC_SEG_TO_DE_IMPORT_PROPERTIES:** Used to map properties between Oracle Unity and Salesforce Marketing Cloud instances.

Key	Description	Example
IsSendable	Stores the flag to identify if the data extension can be used to send marketing emails.	True
IsTestable	Stores the flag to identify if the data extension can be used to test marketing emails.	True
UpdateType	Stores the behavior type of import. For example: AddAndUpdate, AddAndDoNotUpdate, UpdateButDoNotAdd, and OverWrite	AddAndUpdate
InboundDirectory	Stores the location of the FTP directory that stores the files from Oracle Unity.	/Import/Inbound
FileNamePattern	Stores the naming pattern of the file.	*.csv
OutboundDirectory	Stores the location of the FTP directory that stores the files before being imported to Salesforce Marketing Cloud.	/Import
UnityTenantId	Stores the tenant ID of the Oracle Unity instance.	100046
DefaultFieldType	Stores the default field data type in Salesforce Marketing Cloud.	Text
DefaultFieldLength	Stores the maximum field length.	50
AllowErrors	Stores the flag to identify whether to continue the import operation if any error occurs during operation.	True
FieldMappingType	Stores the strategy to map fields from CSV file to the data extension.	InferFromColumnHeadings
FileTransferLocation	Stores the name of the file location created in Salesforce Marketing Cloud.	Name of the file location created in Salesforce Marketing Cloud.

- **ORACLE-BRT-SFMC_SEG_TO_DE_FIELD_TYPE** Used to map the field names in Oracle Unity with field type and field length in Salesforce Marketing Cloud.

FieldName	FieldType	FieldLength
Email	EmailAddress	254
Phone	Phone	50

Activate and Run the Recipe

After you've configured the connections and other resources, activate the recipe package and run it.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration properties. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions** . . . on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
emailSubjectFailure	Specify the subject of the email sent when the import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.
emailFrom	Specify the email address from which run-time exception emails are sent. Enter an email address of your choice.
emailTo	Specify the email address to which run-time exception emails are sent. Enter an email address of your choice.
retryLimit	Specify the number of possible retries if any invoke fails at run time.
emailSubjectSuccess	Specify the subject of the email sent when the import from Oracle Unity to Salesforce Marketing Cloud is successful.

- c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** . . . on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

Note

- You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).
- Ensure that the CSV file is created in the FTP location inbound directory.

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to Salesforce Marketing Cloud and check for the new segments.

- a. Click **Contact Builder**.
- b. Click **Data Extensions**. In the search field, search for new segments created.

Related Documentation

- *Using the Oracle Unity Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Import Supplier Records Asynchronously from Oracle ERP Cloud to Oracle Autonomous Transaction Processing

Use this recipe to import supplier records asynchronously from Oracle ERP Cloud to a PaaS database such as Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to import supplier data asynchronously from Oracle ERP Cloud to Oracle Autonomous Transaction Processing database. To achieve this, a wrapper PL/SQL procedure is created in the database that employs DBMS_SCHEDULER.CREATE_JOB to invoke the main package asynchronously.

There might be situations where PL/SQL packages need to be invoked for data synchronization. Due to the complexity of the logic, these PL/SQL procedures might take 20 minutes or more to execute. As Oracle Integration has a timeout limit of 4 minutes for on-premises databases and 5 minutes for public databases, the integration will fail if the PL/SQL procedure exceeds these time limits. This recipe demonstrates strategies to avoid timing issues and ensure successful data synchronization.

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe.

In a single database package **XX_WRAPPER_SUPPLIERS_PKG**, the following two PL/SQL procedures are written:

1. **RUN_SCHEDULER_JOB**: It accepts the data in a table type input parameter, inserts all the data into the table, and then calls the second package using DBMS_SCHEDULER.CREATE_JOB. This procedure is called from Oracle Integration and accepts all the data in bulk.
2. **PROCESS_SUPPLIERS**: It is being called from the RUN_SCHEDULER_JOB procedure. This is the main PL/SQL procedure that performs all the complex logic on the data that is inserted by the first package, and finally updates the data with the status VALID/INVALID.

A scheduled integration fetches the supplier records in chunks using the Oracle ERP Cloud Adapter and calls the RUN_SCHEDULER_JOB PL/SQL procedure. The invocation does not wait for the response and moves to the next iteration to get another chunk of supplier records. This works until all the supplier records are inserted into the database asynchronously.

System and Access Requirements

- Oracle Integration 3
- Oracle Fusion Cloud
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Autonomous Transaction Processing and Oracle ERP Cloud instances to successfully connect to these external systems using Oracle Integration and import the supplier records asynchronously.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Oracle Autonomous Transaction Processing

Log in to your Oracle Autonomous Transaction Processing database instance as an **Administrator** and perform the following tasks.

1. Perform general configuration tasks. See *Prerequisites for Creating a Connection in Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.
2. Create necessary objects.

To create necessary objects, run the following script:

```
CREATE OR REPLACE EDITIONABLE TYPE SUPPLIER_RECORD AS OBJECT (  
  Supplier VARCHAR2(200),  
  SupplierNumber VARCHAR2(200),  
  TaxOrganizationTypeCode VARCHAR2(200),  
  TaxOrganizationType VARCHAR2(200),  
  SupplierTypeCode VARCHAR2(200),  
  SupplierStatus VARCHAR2(200),  
  BusinessRelationshipCode VARCHAR2(200),  
  BusinessRelationship VARCHAR2(200),  
  TaxRegistrationCountry VARCHAR2(200),  
  TaxpayerCountryCode VARCHAR2(200),
```

```
TaxpayerCountry VARCHAR2(200),
TaxpayerId VARCHAR2(200)
);
/
CREATE OR REPLACE EDITIONABLE TYPE SUPPLIER_TABLE IS TABLE OF
supplier_record;
/
CREATE SEQUENCE SUPPLIERS_SEQ MINVALUE 1 MAXVALUE
999999999999999999999999999999 INCREMENT
BY 1 START WITH 6537 CACHE 20 NOORDER NOCYCLE NOKEEP NOSCALE GLOBAL ;
CREATE TABLE XX_SUPPLIERS
( "ID" NUMBER,
"INSTANCE_ID" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"STATUS" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"SUPPLIER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"SUPPLIER_NUMBER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"TAX_ORGANIZATION_TYPE_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"TAX_ORGANIZATION_TYPE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"SUPPLIER_TYPE_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"SUPPLIER_STATUS" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"BUSINESS_RELATIONSHIP_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"BUSINESS_RELATIONSHIP" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"TAX_REGISTRATION_COUNTRY" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"TAXPAYER_COUNTRY_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"TAXPAYER_COUNTRY" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"TAXPAYER_ID" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
"CREATION_DATE" TIMESTAMP (6) DEFAULT SYSDATE,
"UPDATION_DATE" TIMESTAMP (6) DEFAULT SYSDATE
) DEFAULT COLLATION "USING_NLS_COMP" ;
CREATE UNIQUE INDEX XX_SUPPLIERS_PK ON XX_SUPPLIERS ("ID")
;
ALTER TABLE XX_SUPPLIERS ADD CONSTRAINT "XX_SUPPLIERS_PK" PRIMARY KEY
("ID")
USING INDEX XX_SUPPLIERS_PK ENABLE;
CREATE OR REPLACE EDITIONABLE TRIGGER XX_SUPPLIERS_TRIGGER
BEFORE INSERT ON XX_SUPPLIERS
FOR EACH ROW
WHEN (new.ID IS NULL) BEGIN
:new.ID := SUPPLIERS_SEQ.NEXTVAL;
END;
/
ALTER TRIGGER XX_SUPPLIERS_TRIGGER ENABLE;
CREATE OR REPLACE EDITIONABLE TRIGGER XX_SUPPLIERS_UPDATE_DATE BEFORE
UPDATE ON
XX_SUPPLIERS
FOR EACH ROW
BEGIN
:new.UPDATION_DATE := SYSDATE;
END;
/
ALTER TRIGGER XX_SUPPLIERS_UPDATE_DATE ENABLE;
/
CREATE OR REPLACE EDITIONABLE PACKAGE "ADMIN"."XX_WRAPPER_SUPPLIERS_PKG" AS
PROCEDURE RUN_SCHEDULER_JOB (
P_SUPPLIER_RECORD IN SUPPLIER_TABLE,
P_INSTANCEID IN VARCHAR2
```

```

);
PROCEDURE PROCESS_SUPPLIERS (
P_INSTANCEID IN VARCHAR2
);
PROCEDURE GET_SUPPLIERS_PAGINATION_PROC (
I_PAGE IN NUMBER,
I_LIMIT IN NUMBER,
I_PAGE_SIZE OUT NUMBER,
P_DATA OUT SYS_REFCURSOR
);
END XX_WRAPPER_SUPPLIERS_PKG;
/
CREATE OR REPLACE EDITIONABLE PACKAGE BODY
"ADMIN"."XX_WRAPPER_SUPPLIERS_PKG" AS
PROCEDURE RUN_SCHEDULER_JOB (
P_SUPPLIER_RECORD IN SUPPLIER_TABLE,
P_INSTANCEID IN VARCHAR2
) IS
L_JOB_ACTION VARCHAR2(1000);
BEGIN
-- DBMS_SESSION.sleep(360);
FOR I IN 1..P_SUPPLIER_RECORD.COUNT LOOP
INSERT INTO XX_SUPPLIERS (
INSTANCE_ID,
STATUS,
SUPPLIER,
SUPPLIER_NUMBER,
TAX_ORGANIZATION_TYPE_CODE,
TAX_ORGANIZATION_TYPE,
SUPPLIER_TYPE_CODE,
SUPPLIER_STATUS,
BUSINESS_RELATIONSHIP_CODE,
BUSINESS_RELATIONSHIP,
TAX_REGISTRATION_COUNTRY,
TAXPAYER_COUNTRY_CODE,
TAXPAYER_COUNTRY,
TAXPAYER_ID
) VALUES ( P_INSTANCEID,
'IN-PROGRESS',
P_SUPPLIER_RECORD(I).SUPPLIER,
P_SUPPLIER_RECORD(I).SUPPLIERNUMBER,
P_SUPPLIER_RECORD(I).TAXORGANIZATIONTYPECODE,
P_SUPPLIER_RECORD(I).TAXORGANIZATIONTYPE,
P_SUPPLIER_RECORD(I).SUPPLIERTYPECODE,
P_SUPPLIER_RECORD(I).SUPPLIERSTATUS,
P_SUPPLIER_RECORD(I).BUSINESSRELATIONSHIPCODE,
P_SUPPLIER_RECORD(I).BUSINESSRELATIONSHIP,
P_SUPPLIER_RECORD(I).TAXREGISTRATIONCOUNTRY,
P_SUPPLIER_RECORD(I).TAXPAYERCOUNTRYCODE,
P_SUPPLIER_RECORD(I).TAXPAYERCOUNTRY,
P_SUPPLIER_RECORD(I).TAXPAYERID );
END LOOP;
L_JOB_ACTION := 'DECLARE
BEGIN
XX_WRAPPER_SUPPLIERS_PKG.PROCESS_SUPPLIERS('' '
|| P_INSTANCEID

```

```

|| ''');
END;';
DBMS_OUTPUT.PUT_LINE('JOB Action ' || L_JOB_ACTION);
DBMS_SCHEDULER.CREATE_JOB(JOB_NAME => 'supplier_job', JOB_TYPE =>
'PLSQL_BLOCK',
JOB_ACTION => L_JOB_ACTION, START_DATE => SYSTIMESTAMP
, AUTO_DROP => TRUE,
ENABLED => TRUE);
END;
PROCEDURE PROCESS_SUPPLIERS (
P_INSTANCEID IN VARCHAR2
) IS
BEGIN
-- write your logic here, get all data based on Instance Id and do the
validation and mark data as VALID OR INVALID
UPDATE XX_SUPPLIERS
SET
STATUS = 'VALID'
WHERE
INSTANCE_ID = P_INSTANCEID;
END;
PROCEDURE GET_SUPPLIERS_PAGINATION_PROC (
I_PAGE IN NUMBER,
I_LIMIT IN NUMBER,
I_PAGE_SIZE OUT NUMBER,
P_DATA OUT SYS_REFCURSOR
) AS
COUNT1 NUMBER;
BEGIN
SELECT
COUNT(*)
INTO COUNT1
FROM
SUPPLIERS;
I_PAGE_SIZE := CEIL(COUNT1 / I_LIMIT);
OPEN P_DATA FOR SELECT
*
FROM
SUPPLIERS
ORDER BY
ID
OFFSET NVL((I_PAGE - 1), 0) * I_LIMIT ROWS FETCH NEXT I_LIMIT ROWS
ONLY;
END;
END XX_WRAPPER_SUPPLIERS_PKG;
/

```

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Autonomous Transaction Processing Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See [Configure Oracle Autonomous Transaction Processing](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Select the check box, then click Upload to upload the wallet file. See Configure Oracle Autonomous Transaction Processing .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle ERP to DB Supplier Synch** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify that the supplier data is written to the ATP database table.

Related Documentation

- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)
- [Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3](#)

Import Supplier Records from SAP S/4HANA to Oracle ERP Cloud

Use this recipe to import supplier records from SAP S/4HANA to Oracle ERP Cloud Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports supplier records from SAP S/4HANA to Oracle ERP Cloud according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the supplier records from the SAP S/4HANA instance and, in turn, creates corresponding records in your Oracle ERP Cloud instance. SAP S/4HANA supplier records

along with their associated data (such as name, alternate name, tax organization type, and business relationship type) are imported to Oracle ERP Cloud.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- SAP S/4HANA
- An account on SAP S/4HANA with the Administrator role
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SAP S/4HANA and Oracle ERP Cloud instances in order to successfully connect to these external systems using Oracle Integration and transfer supplier records between them.

Configure SAP S/4HANA

To configure SAP S/4HANA, see Prerequisites for Creating a Connection in *Using the SAP S/4HANA Cloud Adapter with Oracle Integration 3*.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the SAP S/4HANA Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the host name of the SAP S/4HANA instance.
3. In the Security section, enter the username and password of the Communication User created in the SAP S/4HANA instance. See [Configure SAP S/4HANA](#).
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table to map SAP S/4HANA data fields to Oracle ERP Cloud fields. In addition, enter an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. In the **Property** column, enter the field names of SAP S/4HANA.
3. In the **Value** column, enter the corresponding field names of Oracle ERP Cloud, which are to be mapped to the SAP S/4HANA fields.
4. In the last row, enter **EmailTo** in the **Property** column, and enter an email ID of your choice in the **Value** column.
5. Click **Save**. If prompted, click **Save** again.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, you can specify a value for the `lastRun` parameter, and click **Run**.

The `lastRun` parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the SAP S/4HANA supplier records created 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. To fetch all the supplier records from the beginning, set the value as zero. You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle ERP Cloud instance and check for the new suppliers created.
 - a. On the home page, click **Menu**.
 - b. From the list, under the Procurement section, click **Suppliers**.

The list of suppliers is displayed, which contains the records imported from SAP S/4HANA.

Related Documentation

- [Using the SAP S/4HANA Cloud Adapter with Oracle Integration 3](#)
- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)

Import Suppliers from File Server to Oracle ERP Cloud

Use this recipe to import suppliers from File Server to Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports an interface file containing supplier information from File Server to Oracle ERP Cloud according to a schedule specified in Oracle Integration.

In this recipe, File Server is used to store the supplier file and a native **File Server** action is used to list and read the file.

This recipe contains the following integration flows:

1. **Oracle FTP ERP Import Suppliers:** This is a schedule-based integration flow that reads the file from File Server, does the conversion as per the suppliers FBDI file and imports it into Oracle ERP Cloud using the Oracle ERP Cloud adapter.
2. **Suppliers Callback:** This is an event-based integration flow that will be called once the Import Suppliers job is completed in Oracle ERP Cloud. This flow uploads the log file to File Server.

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe.

To use File Server, which is an embedded sFTP server within Oracle Integration, you must enable it from your Oracle Integration instance. See *Enable File Server in Using File Server in Oracle Integration 3*.

You can also use an external FTP server instead of File Server but you would need to configure the FTP adapter connection if you are using an external FTP server.

System and Access Requirements

- Oracle Integration 3
- Oracle ERP Cloud
- An account on Oracle ERP Cloud with the Administrator role
- File Server

Before You Install the Recipe

You must perform the following configuration tasks on your File Server and Oracle ERP Cloud instances in order to connect to these external systems using Oracle Integration, and successfully import suppliers.

Configure File Server

You have to create the required directories in File Server.

1. Create a directory to read the files. Note down the path of the directory. For example: /home/user2/source.
2. Create a directory to archive files that have been processed. Note down the path of the directory. For example: /home/user2/archive.
3. Create a directory to write the log file. For example: /home/user2/logs.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **Integration_Lookup** lookup table to specify the file server directories.

1. In the Lookups section, click the lookup name.
2. Enter appropriate values in the **Property_value** column for the following properties.

Property_name	Property_value
SourceDir	Enter the path of the directory on File Server from where the file must be read. See Configure File Server . For example: /home/user2/source.
ArchiveDir	Enter the path of the directory on File Server to archive the file. For example: /home/user2/archive.
LogFileDir	Enter the path of the directory on File Server to write the log file. For example: /home/user2/logs.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle FTP ERP Import Suppliers** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to Oracle ERP Cloud and verify that new suppliers are created.
5. Go to the directory that contains the log file on File Server and verify that the zip file containing various log files (indicating the status of the suppliers) has been created.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using File Server in Oracle Integration 3*

Import Users from Oracle HCM Cloud to Okta

Use this recipe to create Okta users based on new hire information in Oracle HCM Cloud.

Topics:

- [About This Recipe](#)

- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Okta instances in order to successfully connect to these external systems using Oracle Integration and import users.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and import users, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure Okta API Key

To access the Okta instance from Oracle Integration, you'll require the API key associated with your Okta account.

To obtain your Okta account's API key:

1. Log in to your Okta instance as an administrator.
2. Click **Admin** to open the Admin console.
3. From the menu, select **Security** and choose **API**.
4. Navigate to the **Tokens** tab.
5. Note the API Token.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

Use the following steps to configure the **Oracle HCM Cloud Connection** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Okta Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Specify your Okta URL. For example: <code>example.okta.com</code> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave API Key Based Authentication selected.
API Key	Enter the API key associated with your Okta account. See Configure Okta API Key .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** . . . on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Okta instance and check for new users.

To view the list of users in Okta, navigate to **Directory** and select **People**.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Import Worker Data from an FTP Server to Oracle HCM Cloud

Use this recipe to import worker data from an FTP server to Oracle HCM Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe imports worker data from an FTP server to Oracle HCM Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When triggered, the recipe's integration flow picks a comma-separated values (CSV) file that contains the worker data from a specified location in an FTP server. The imported CSV file is converted to NXSD format in Oracle Integration. The NXSD is used within Oracle Integration to generate a DAT file that is compatible with HCM Data Loader. The DAT file with the worker data is then uploaded to Oracle HCM Cloud using HCM Data Loader.

Note that for successfully processing the worker data in Oracle Integration, the input file should be in a particular format. You can download a sample input file `worker.csv` from [Sample file](#).

System and Access Requirements

- Oracle Integration, Version 23.06.02 or higher
- Oracle HCM Cloud
- An account on Oracle HCM Cloud with the Administrator role
- A secure FTP (sFTP) server
- An FTP client to access the sFTP server

Before You Install the Recipe

You must perform the following configuration tasks on your FTP server and Oracle HCM Cloud instance in order to connect to these external systems using Oracle Integration, and successfully import worker data.

Access Your FTP Server

Obtain an sFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an input folder. Note down the path of this folder.
3. Upload the CSV file (Worker .csv) with the worker records into the input folder.

Configure Oracle HCM Cloud

To access Oracle HCM Cloud using Oracle Integration and import worker data, you must perform certain configurations on your Oracle HCM Cloud instance.

Log in to your Oracle HCM Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Note the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle HCM Cloud from Oracle Integration.
2. Enable the NXSD file generation feature.

This feature determines whether you can view the **Generate Mapping Files** parameter when you submit files for bulk data loading and whether mapping files are generated when you refresh business objects.

- a. Navigate to **Setup and Maintenance**.
 - b. From the **Functional Areas** menu, select **HCM Data Loader**.
 - c. From the HCM Data Loader tasks, select **Configure HCM Data Loader**.
 - d. In the Configure HCM Data Loader window, under **Availability** category, select the **Generate Business Object Mapping Files** parameter.
 - e. Set the **Override** value of the parameter to **Yes**.
 - f. Click **Save and Close**.
3. Generate an NXSD file within HCM Data Loader.

To generate an NXSD file for a specific business object for the first time:

- a. In the Oracle HCM Cloud navigator menu, under the Data Exchange section, click **HCM Data Loader**.
- b. In HCM Data Loader, click **View Business Objects**.
- c. Select the specific business object that you wish to generate an NXSD template for.
- d. Click **Refresh Object**.

An integration mapping file is generated and downloaded to your local system. Additionally, a copy of this file is also placed in Oracle WebCenter Content.

Note

If changes have occurred to the business object after the NXSD was downloaded, the above steps can be repeated to generate a new version.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your FTP server.
FTP Server Port	Enter the port to connect to your FTP server.
SFTP Connection	Select Yes or No based on your FTP server protocol.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the integration flow. You can update the integration properties only after activating the integration flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the following integration properties with appropriate values.
 - **InputFolder**: Enter the path of the input folder created in FTP server. See [Access Your FTP Server](#).
 - **FromAddress**: Enter the email address from which notification emails have to be sent.
 - **ToAddress**: Enter the email address to which notifications have to be sent in case of errors.
 - c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Oracle HCM Cloud instance and check if the sample worker data have been imported.

Related Documentation

- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*

Increment Variable in a Loop

Use this recipe to increment the sequence or counter variable within a data mapper while processing data in batches.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe uses a REST adapter to fetch or receive data in batches from an external application. The recipe enables you to increment the sequence or counter variable within a data mapper, using the `position()` function, for each batch of data that's processed.

To use the recipe, you must install the recipe and configure the connections within it. Subsequently, you can activate the integration flow of the recipe and send a REST request with the data to the integration's endpoint URL from an external application. When triggered, the recipe's integration flow returns a response to the external application and also logs the request payload in Oracle Integration, which you can view in the activity stream.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Connection

1. In the Connections section, click the connection name.

2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the data. Provide the data in the POST request's **Body** field. See the following step for sample data.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click **Body** and enter sample data.
For example:

```
{"accounting": [ { "firstName": "John1", "lastName": "Doe1", "age": 23 } , { "firstName": "John2", "lastName": "Doe2", "age": 33 } , { "firstName": "John3", "lastName": "Doe3", "age": 44 } , { "firstName": "John4", "lastName": "Doe4", "age": 53 } , { "firstName": "John5", "lastName": "Doe5", "age": 63 } ] }
```
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the test page, you'll find a success status 200 OK, with the response returned.
4. View the response in the Activity stream pane. Click **View payload**  on the last item of the activity stream to view the response returned.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*

Integrate DiCentral with Oracle SCM Cloud

Use this recipe to create forecast and sales orders in Oracle Fusion Cloud Supply Chain & Manufacturing (Oracle SCM Cloud) for original equipment manufacturer (OEM) initiated Electronic Data Interchange (EDI) releases from DiCentral.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe integrates Oracle SCM Cloud with DiCentral (a third-party application). It uses DiCentral's OEM initiated EDI capabilities to gather forecast and firm order information, and complements it with Oracle SCM Cloud's supply chain management capabilities to provide release cumulative management.

Note

This is a package-based recipe. See About Packages and Edit and Replace Dependent Resources in a Package

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. When forecast and sales order information is generated in DiCentral, a REST request is sent to Oracle Integration with the necessary data to create the corresponding forecast and sales order in Oracle SCM Cloud. Also, when the shipment and invoice for the sales order is generated in Oracle SCM Cloud, Oracle Integration sends this information back to DiCentral. Additionally, the sales order creation response is also sent back to DiCentral.

The recipe package contains four integration flows. Initially a REST request (with necessary data) from DiCentral triggers the following two integration flows of the recipe.

- **Oracle DiCentral SCM Forecast Create:** This flow receives the EDI forecast from DiCentral and sends the forecast data to Oracle SCM Cloud for forecast creation.
- **Oracle DiCentral SCM SalesOrder Create:** This flow receives EDI firm orders from DiCentral and invokes Oracle SCM Cloud to create sales orders. Upon successful order creation, response is sent back to DiCentral.

When the shipment and invoice are fulfilled on the sales orders in Oracle SCM Cloud, the following two integration flows of the recipe are triggered:

- **Oracle SCM DiCentral Order Shipment:** This flow receives the shipment information from Oracle SCM Cloud and sends it to DiCentral. When a shipment is created, Collaborative Messaging Framework (CMK) is used to trigger this integration from Oracle SCM Cloud.
- **Oracle ERP DiCentral Order Invoice:** This integration flow receives the invoice information from Oracle ERP Cloud when invoice is created for successful shipments and sends it to DiCentral. When an invoice is created, CMK is used to trigger this integration from Oracle ERP Cloud.

System and Access Requirements

- Oracle Integration, Version 22.1.3.0.0 or higher
- An account on Oracle Integration with the Service Invoker role.
- Oracle SCM Cloud and Oracle ERP Cloud, Version 21D or higher
- An account on Oracle SCM Cloud and Oracle ERP Cloud with the Administrator role.
- DiCentral

Assumptions

You have installed the **REST — Oracle Fusion Applications | Invoke Pop-Up Notifications** recipe on your Oracle Integration instance.

You'll require this recipe to send the notification information to Oracle SCM Cloud.

To install this recipe, see [Invoke Pop-Up Notifications for Events in Oracle Fusion Applications](#).

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle SCM Cloud and DiCentral instances to successfully run the recipe.

Configure Oracle SCM Cloud

To access Oracle SCM Cloud from Oracle Integration, create forecast and sales orders, and send back sales order response as well as shipment & invoice response, you'll need to perform the following configuration tasks on your Oracle SCM Cloud instance.

1. Perform all the base configurations in Oracle SCM Cloud, including setting up security, access roles, inventory, order management, receivables and supply chain planning setups.
2. Configure order management fulfillment line and header extensible flex fields to capture OEM firm order reference information. Receivables invoice lines DFF also need to be configured to maintain the OEM firm order references so that the same information can be sent back to DiCentral via the integration.
 - a. Refer to Order management to AR Mapping as per support note 2446740.1.
 - b. Configure the Collaboration Messaging Framework (CMK) to establish B2B message exchange for sending shipment and invoice information to DiCentral.
 - c. Create a reference/template sales order with a custom order type to reference sales order information.
3. Create a custom editable measure EDI Shipments Forecast to receive forecast data from DiCentral.
4. Create a user account for Oracle Integration on Oracle SCM Cloud.

Log in to the Oracle SCM Cloud instance as an **Administrator** and perform the following tasks.

- a. To access Oracle SCM Cloud from Oracle Integration, you must create a user account for Oracle Integration. Note the user name and password of the account, as you'll use these credentials to connect to Oracle SCM Cloud from Oracle Integration, using the Oracle ERP Cloud Adapter.

- b. Assign the required roles to the user account. See *Assign Required Roles to an Integration User* in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
- c. In addition to the above roles, also assign the following roles and privileges to the account:
 - Accounts Receivable Manager
 - Demand and Supply Planner
 - Employee
 - IT Security Manager
 - Human Capital Management Integration Specialist
 - Order Manager
 - Sales Administrator
 - Shipping Manager
 - Integration Specialist
 - View_All_HCM_Integration_Specialist
 - AttachmentsUser
 - SOAOperator
 - FND_MANAGE_CATALOG_SERVICE_PRIV
 - Customer Relationship Management Application Administrator
5. Obtain the Collaboration Messaging Framework (CMK) WSDL. See [Overview of Collaboration Message Definitions](#) in *Supply Chain and Manufacturing* documentation.

Note that you have to download the CollaborationMessageV2Service WSDL.

Configure DiCentral

To access DiCentral from Oracle Integration, you'll need to perform the following configuration tasks on your DiCentral instance.

1. Configure the outbound payloads from DiCentral.

To configure the outbound payloads, you'll require the following information from the Oracle SCM Cloud instance. This will be cross referenced from the request coming in from DiCentral to Oracle SCM Cloud.

 - Customer
 - Customer Product Number
 - Product Number
 - Ordered UOM Code
 - Bill to Party Number
 - Bill to Site Identifier
 - Ship to Party Number
 - Ship to Site Identifier
 - Supplier Organization Code

2. Obtain the endpoint URL for Sales Order response, and Shipment & Invoice response to be sent to DiCentral from Oracle Integration. You'll require this while configuring connection from Oracle Integration.
3. Obtain the API key associated with your Dicentral instance. You'll require this while configuring connection from Oracle Integration.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources. See [Configure Resources in a Package](#).

Configure the REST OIC API Connection

1. In the Configuration Editor, under **Connections**, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the host name of your Oracle Integration instance.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of the Oracle Integration user account with the Service Invoker role.
Password	Enter the password of the Oracle Integration user account with the Service Invoker role.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the Configuration Editor, click **Go back** .

Configure the Oracle ERP Cloud DiCentral Connection

1. In the Configuration Editor, under **Connections**, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .

Field	Information to Enter
Username	Enter the username of the account created for Oracle Integration on Oracle SCM Cloud. See Configure Oracle SCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle SCM Cloud.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the Configuration Editor, click **Go back** .

Configure the REST DiCentral Sales Order Connection

- In the Configuration Editor, under **Connections**, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the Sales Order Response endpoint you obtained from DiCentral. See Configure DiCentral .

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select API Key Based Authentication .
API Key	Enter the API key you obtained from DiCentral. See Configure DiCentral .
API Key Usage	Ensure that the following is configured: -H Authorization: Bearer \${api-key}

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the Configuration Editor, click **Go back** .

Configure the REST DiCentral Shipment Invoice Connection

- In the Configuration Editor, under **Connections**, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the Sales Order Response endpoint you obtained from DiCentral. See Configure DiCentral .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select API Key Based Authentication .
API Key	Enter the API key you obtained from DiCentral. See Configure DiCentral .
API Key Usage	Ensure that the following is configured: -H Authorization: Bearer \${api-key}

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the Configuration Editor, click **Go back** .

Configure the REST OIC API Connection

1. In the Configuration Editor, under **Connections**, click the connection name.
2. In the **WSDL URL** field of the Connection Properties section, upload the WSDL file that you downloaded from your Oracle SCM Cloud instance. See [Configure Oracle SCM Cloud](#).
3. In the **Security** field, select **Basic Authentication**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the Configuration Editor, click **Go back** .

Configure the Lookup Table

Edit the lookup table, and enter the necessary values.

1. In the Configuration Editor, click **Lookups** to view the lookup tables in the recipe. Click the name of the lookup table to edit.
2. This recipe package contains the following lookup table: **Oracle-BRT-SCM_DiCentral_Configuration**. This lookup table contains the information needed to create a sales order in Oracle SCM Cloud.

The following table provides the description for each parameter:

Name	Description
SourceTransactionSystem	Transaction source system identify the source of transactions that you import into Oracle SCM Cloud. Example value: DIC Note Update this value based on the configuration in Oracle SCM Cloud.
MeasureName	Measure name to import sales forecasts to Oracle ERP Cloud. Example value: EDI Shipments Forecast Note Do not modify this value unless the Oracle SCM Cloud configuration has changed.
Currency	Currency used for sales order transaction. Example value: USD Note Update this value based on the configuration in Oracle SCM Cloud.
DeletePreviousForecast	Flag to delete the previous imported forecasts. Example value: No Note Do not modify this value.
EmailNotification	Notify the end user if any error occurs in the integration process. Note Update based on who needs to receive error notification.

Name	Description
OrderTemplateName	A sales order to provide the value for sales order transaction in Oracle SCM Cloud. Example value: DIC_TEMPLATE_ORDER Note Update this value based on the configuration in Oracle SCM Cloud.
EffContextCodeVO	The VO ContextCode defined for FlexFields as per Oracle SCM Cloud. Example value: FulfillLineEffBEDI_Release_InformationprivateVO Note Update this value based on the configuration in Oracle SCM Cloud.
SalesOrder_Notification_User	User name of the fusion user who receives the Fusion pop-up notifications. This value is case sensitive. Example value: Robert.James Note Update this value based on who needs to receive notification in Oracle SCM Cloud for successful sales order creation.
SalesForecasts_Notification_User	User name of the fusion user who receives the Fusion pop-up notifications. This value is case sensitive. Example value: jack.wells Note Update this value based on who needs to receive notification in Oracle SCM Cloud for successful forecast creation.
TaskCreatorForNotificationUser	Name of the task creator which is displayed in the Fusion pop-up notification. Example value: DiAutomotive Note Update this value based on customer preference.

Name	Description
ContextCodeName	The EFF ContextCode defined for FlexFields as per Fusion configuration. Example value: EDI Release Information

 Note

Update this value based on the configuration in Oracle SCM Cloud.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the Configuration Editor, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Obtain the endpoint URLs for the trigger integration flows, namely **Oracle DiCentral SCM Forecast Create** and **Oracle DiCentral SCM SalesOrder Create**.
 - i. On the Configuration Editor page of Oracle Integration, select an integration flow and click **Run**.
 - ii. On the resulting pop-up dialog, click the **Metadata URL** link.
 - iii. From the new tab that opens, copy the **Endpoint URL** value.
 - b. Log in to your DiCentral instance and configure it to perform the following actions to send forecast and firm orders.
 - i. Send a POST EDI forecast request to the **Oracle DiCentral SCM Forecast Create** endpoint URL (obtained in the previous step) along with the forecast data.
 - ii. Send a POST EDI firm order request to the **Oracle DiCentral SCM SalesOrder Create** endpoint URL (obtained in the previous step) along with the firm order data.

When the recipe receives these POST requests it will send the forecast or firm order requests to Oracle SCM Cloud.

- c. Obtain the endpoint URLs for the integration flows, namely **Oracle SCM DiCentral Order Shipment** and **Oracle ERP DiCentral Order Invoice**.
 - i. On the Configuration Editor page, select an integration flow and click **Run**.
 - ii. On the resulting pop-up dialog, click the **WSDL URL** link.
 - iii. From the new tab that opens, copy the **WSDL URL** value. Leave out **?wsdl** from the URL, and note the URL.
- d. Log in to Fusion Applications, and configure the shipment and invoice endpoints in Fusion CMK, so that when a shipment or invoice is created in Oracle SCM Cloud, the recipe sends the information to DiCentral.

- i. Navigate to **Collaboration Messaging** under **Tools**. Click **Tasks**, then click **Service Providers**, and then select **Manage Collaboration Messaging Service Providers**.
- ii. Open the service provider that you had set up as part of Fusion CMK configuration.

Note

If not already configured, you can configure Web Services as delivery method for both shipment and invoice outbound CMK messages.

- iii. Update the service provider using the WSDL URLs you have obtained in the previous step.

Update Shipment Provider:

Field	Information to enter
Name	Name of your Shipment Out delivery method. For example, SHIPMENT_OUT.
Delivery Method Type	Web Service
Service Name	CollaborationMessageV2.process
End Point	Endpoint URL of the shipment webservice.
User Name	Enter the username of the Oracle Integration user account with the Service Invoker role.
Password	Enter the password of the Oracle Integration user account with the Service Invoker role.

Update Invoice Provider:

Field	Information to enter
Name	Name of your Invoice Out delivery method. For example, INVOICE_OUT.
Delivery Method Type	Web Service
Service Name	CollaborationMessageV2.process
End Point	Endpoint URL of the shipment webservice.
User Name	Enter the username of the Oracle Integration user account with the Service Invoker role.
Password	Enter the password of the Oracle Integration user account with the Service Invoker role.

- e. Log in to your Oracle SCM Cloud as **Supply Chain Planner** and check if the forecast is created.
 - i. Click the **Tools** tab on the home page, and then select **Scheduled Processes**. Ensure that the collection requests have been completed.
 - ii. Click **Supply Chain Planning**, and then select **Demand and Supply Planning**.
 - iii. Open the plan that you created and run the plan with the following options.
 - Refresh with Current Data: Yes
 - Forecast Shipments: Yes
 - iv. When the execution is complete, you can review the plan including exceptions.

- f. Log in to Oracle SCM Cloud as the **Order and Shipping Clerk**, and check for the new order. After the order is fulfilled, ship the order fulfilment line and generate the invoice.
 - i. Check if a new sales order is created and view its details.
 - Click the **Bell** icon on the top-right corner to view the order creation notifications.
 - Copy the Document Reference Number from the notification message.
 - Search for the new order with the Document Reference Number. The new order is displayed with information such as customer name, quantity, and so on. Note the order number. You'll use it later to ship the order.
 - Click **Additional Information** in the header to view the OEM specific firm order information received from DiCentral.
 - Select the fulfillment line with earliest shipment date. Observe the order status. It should show *Awaiting Shipping*.
 - Click **Additional Information** on the right for the selected fulfilment line and view the additional flex fields.
 - ii. To ship the selected fulfilment line, click **Supply Chain Management** and then click **Inventory Management**.
 - Select **Show Tasks: Shipments** and search for the order with the order number you noted down earlier.
 - Select the fulfillment line with earliest ship date. Enter the same Shipped Quantity as Requested Quantity and click **Save**.
- Note**

The recipe doesn't support split lines.
- Click **AutoCreate Shipment**.
 - Click the shipment you just created. Click **Actions**, and then click **Pick Release**. Click **Actions** again and click **Pack Automatically**.
 - Click **Ship Confirm**.
- When the order is shipped, the integration **Oracle SCM DiCentral Order Shipment** is triggered, and it sends the shipment information to DiCentral.
- iii. To generate an invoice for the shipment, click the **Tools** tab on the home page, and then select **Scheduled Processes**.
 - Schedule a new process for **Import AutoInvoice**. Run the process for the order you shipped. Once the process completes, an invoice is generated.

When the invoicing process completes, the integration **Oracle ERP DiCentral Order Invoice** is triggered, and it sends the invoice information to DiCentral.
 - iv. If the request from DiCentral contains updates to existing orders, you should see the updates in the order. Updates could be for the quantity or dates.
3. Monitor the running of the integration flow in Oracle Integration.
 - a. In the Configuration Editor, click **Activation** to view the integrations in the recipe. Hover over the integration to monitor, then click **Actions** ... and select **Run**, then **Track instances**.

- b. On the Track Instances page, observe the integration flow of the recipe being triggered and running successfully.
 4. Finally, log in to your DiCentral instance and verify if the order creation response, and the shipment and invoice information have been received.
 - a. To verify if the order creation response is received:
 - Navigate to **Customer Demand Mgt** and then **Firm Orders**.
 - Filter the records using trading partner you have used for sending the request to Oracle SCM Cloud. The orders created with Oracle SCM Cloud are displayed. The External Reference Number shows the order key for each order created in Oracle SCM Cloud.
 - Search for the required record and click it to view details. Check if the record details match with the order created in Oracle SCM Cloud.
 - b. To verify if the shipment information is received:
 - Navigate to **Customer Shipments** and then **Shipments**.
 - Filter the records using trading partner you have used for sending the request to Oracle SCM Cloud. Search will display records with individual ship IDs.
 - Click the record for which you want to view details. Check if the shipment record details match with the shipment created in Oracle SCM Cloud.
 - c. To verify if the invoice information is received:
 - Navigate to **Customer Financing** and then **Invoice**.
 - Filter the invoices using trading partner you have used for sending the request to Oracle SCM Cloud. Invoice records are displayed.
 - Click the required invoice record, and view its summary in the **General** tab. Check if the invoice record details match with the invoice generated for the shipment in Oracle SCM Cloud.
To view the comparison between Requested, Shipped, and Invoice Quantity, click the **Details** tab.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Integrate KanbanBOX with Oracle SCM Cloud

Use this recipe to create purchase orders and work orders in Oracle Fusion Cloud Supply Chain & Manufacturing (Oracle SCM Cloud) when Kanban cards are released in KanbanBOX.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe integrates Oracle SCM Cloud with KanbanBOX (a third-party application). It complements existing supply-chain management capabilities of Oracle SCM Cloud with additional Kanban-replenishment capability.

Note

This is a package-based recipe. See [About Packages and Edit and Replace Dependent Resources in a Package](#)

This is an application-driven recipe that creates the necessary purchase orders and work orders in Oracle Fusion Cloud Procurement and Oracle Fusion Cloud Manufacturing (components of Oracle SCM Cloud), respectively, when a Kanban card is released in KanbanBOX. The bin-replenishment requirements in the production line drive the release of Kanban cards in KanbanBOX.

To use the recipe, you must install the recipe package and configure the connections and other resources within the package. And, when you scan a card (or a batch of cards) as *released* in KanbanBOX, a REST request is sent to Oracle Integration with the necessary cards data to create the corresponding orders. For each card in the request, Oracle Integration creates a purchase order (PO) in Oracle Fusion Cloud Procurement or a work order in Oracle Fusion Cloud Manufacturing, based on the type of request from KanbanBOX. Also, Oracle Integration sends the purchase order and work order information, including status changes, back to KanbanBOX.

The recipe package contains seven integration flows for executing this use case. Initially, a REST request (with necessary data) from KanbanBOX triggers the following two integration flows of the recipe:

- **Oracle KanbanBOX ERP PurchaseOrder Create:** This flow creates a purchase requisition along with line items in Oracle Fusion Cloud Procurement; the requisition in turn automatically creates a purchase order for each KanbanBOX card. (Each Kanban card makes up one order line in the purchase order). Subsequently, the **Oracle ERP KanbanBOX PurchaseOrder Status** integration flow listens to the Purchase Order Creation event in Oracle Fusion Cloud Procurement, and propagates the purchase order number to KanbanBOX for each of the Kanban cards (PO lines).
- **Oracle KanbanBOX ERP WorkOrder Create:** This flow calls the Oracle KanbanBOX ERP Process WorkOrder integration flow to create a work order in Oracle Fusion Cloud Manufacturing for each KanbanBOX card. The **Oracle KanbanBOX ERP Process WorkOrder** flow creates the work orders and sends the work order numbers back to KanbanBOX.

When the purchase order and work order are fulfilled, the following two integration flows of the recipe are triggered:

- **Oracle ERP KanbanBOX PurchaseOrder Receipt:** This flow listens to the Purchase Order Receipt event in Oracle Fusion Cloud Procurement, and propagates the receipt status to KanbanBOX for each of the Kanban card receipts.
- **Oracle ERP KanbanBOX WorkOrder Status:** This flow listens to the Work Order Status Change event in Oracle Fusion Cloud Manufacturing, and propagates the Work Order Completion status to KanbanBOX for each of the Kanban cards.

In addition, the **Oracle ERP KanbanBOX ErrorHandler** integration flow of the recipe blocks the respective Kanban card if any failures are encountered during work order or purchase order creation.

System and Access Requirements

- Oracle Integration, Version 22.1.3.0.0 or higher
- An account on Oracle Integration with the Service Invoker role to configure the recipe
- Oracle SCM Cloud, Version 21D or higher
- An account on Oracle SCM Cloud with the Administrator role, and also Procurement Manager and Manufacturing Plant Manager roles (in order to create receipts and close work orders)
- KanbanBOX

Assumptions

You have installed the **REST — Oracle Fusion Applications | Invoke Pop-Up Notifications** recipe on your Oracle Integration instance. You'll require this recipe to send the notification information to Oracle SCM Cloud when work orders and purchase orders are created.

To install the recipe, see [Invoke Pop-Up Notifications for Events in Oracle Fusion Applications](#).

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle SCM Cloud and KanbanBOX instances in order to successfully run the recipe.

Configure Oracle SCM Cloud

To access Oracle SCM Cloud from Oracle Integration and create purchase orders and work orders, you'll need to perform the following configuration tasks on your Oracle Fusion Cloud SCM instance.

1. Perform all the base configurations in Oracle SCM Cloud, including setting up security, access roles, inventory, procurement, manufacturing, and product hub. Here is a list of important base configurations:
 - a. Configure descriptive flex-fields in work order and/or purchase requisition, depending on your implementation requirements, for storing the Kanban card identifier.
 - b. Optionally, configure a distinct sub-inventory for your Kanban-specific inventory.
 - c. Leverage a Blanket Purchase agreement to automatically convert your purchase requisitions for Kanban items into purchase orders.
2. Create a integration user account on Oracle SCM Cloud.

To access Oracle SCM Cloud from Oracle Integration, you'll require a separate user account on Oracle SCM Cloud.

Log in to the Oracle SCM Cloud instance as an **Administrator** and perform the following tasks.

- a. 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 SCM Cloud from Oracle Integration, using the Oracle ERP Cloud adapter.

- b. Assign the following roles to the user account. See Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
- c. In addition to the above roles, also assign the following roles and privileges to the account:
 - Accounts Receivable Manager
 - Buyer
 - BuyerGSE
 - Cost Accountant
 - Employee
 - GSE Role for RFSmartAdvanced Procurement Requester
 - Inventory Manager
 - Manufacturing Engineer
 - Order Manager
 - Procurement Application Administrator
 - Procurement Catalog Administrator
 - Procurement Manager
 - Procurement Preparer
 - Procurement Requester
 - Production Operator
 - Production Supervisor
 - Quality Engineer
 - Receiving Agent
 - Supplier Administrator
 - Supplier Manager
 - Supplier Qualification
 - Warehouse Manager
 - Warehouse Operator
 - Warehouse OperatorEmployee

Configure KanbanBOX

To access KanbanBOX from Oracle Integration, you'll need to perform the following configuration tasks on your KanbanBOX instance.

1. Obtain the API key associated with your KanbanBOX instance.
2. Configure the outbound payloads from KanbanBOX. In order to configure this, you'll require the following information from the Oracle SCM Cloud instance.
 - Unit of Measurement
 - Customer Number
 - Business Unit
 - Requestor

- Buyer
- Delivery Location
- Organization
- Part Number
- Customer Name
- Supplier Name
- Supplier Number

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources. See [Configure Resources in a Package](#).

Configure the REST OIC API Connection

1. In the Configuration Editor, under **Connections**, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the host name of the Oracle Integration instance in which you've installed the recipe.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the user name of the Oracle Integration user account with the Service Invoker role.
Password	Enter the password of the Oracle Integration user account with the Service Invoker role.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the Configuration Editor, click **Go back** .

Configure the REST KanbanBOX Connection

1. In the Configuration Editor, under **Connections**, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the host name of your KanbanBOX instance.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select API Key Based Authentication .
API Key	Enter the API key obtained from the KanbanBOX instance. See Configure KanbanBOX .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the Configuration Editor, click **Go back** .

Configure the ERP Cloud Connection KanbanBOX

1. In the Configuration Editor, under **Connections**, click the connection name.
2. In the Properties section, enter the Oracle SCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on Oracle SCM Cloud. See Configure Oracle SCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle SCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the Configuration Editor, click **Go back** .

Configure the Lookup Tables

Edit the lookup tables and enter the necessary values.

1. In the Configuration Editor, click **Lookups** to view the lookup tables in the recipe. Click the name of the lookup table to edit.
2. This recipe package contains the following lookup tables.

Oracle-BRT-ERP_KanbanBOX_WO_Config

This lookup table contains the information needed to create a work order in Oracle Fusion Cloud Manufacturing. The following table provides the description for each parameter:

Name	Description
WorkDefinitionCode	The value of the Work Definition code used in the work order creation as configured on the Oracle Fusion Cloud Manufacturing instance. Example value: ORA_MAIN Note Do not modify this value unless the Oracle Fusion Cloud Manufacturing configuration has changed.
WorkOrderStatusCode	The status of the work order created using the integration, which is predefined. Example value: ORA_RELEASED Note Do not modify this value unless the Oracle Fusion Cloud Manufacturing configuration has changed.
ExplosionFlag	The value of the Explosion Flag set while creating the work order as configured on Oracle Fusion Cloud Manufacturing. Example value: TRUE Note Do not modify this value.
KanbanAdmin	The Kanban Admin email ID that must be notified if there are errors while updating KanbanBOX. Example value: abc@xyz.com Update this value based on your KanbanBOX configuration.
WONotificationUserName	The user name of the Oracle Fusion Cloud Manufacturing user who receives the Fusion pop-up notifications on work order creation; this value is case -sensitive. Example value: Terry.Johnson Update this value based on the configuration in Oracle Fusion Cloud Manufacturing.
TaskCreator	The name of the task creator displayed in the Fusion pop-up notifications for work orders. Example value: Terry Johnson Update this value based on the configuration in Oracle Fusion Cloud Manufacturing.

Name	Description
BatchCardLimit	The threshold limit on the number of batch cards to be processed by Oracle Integration in a single Kanban release (via an integration call). Any incoming request having batch cards beyond this number is not processed by Oracle Integration. Example value: 5 Update this value based on the configuration in Oracle Integration.
OICAdminEmail	The Oracle Integration Admin email ID that must be notified on failures. Example value: abc@xyz.com Update this value based on the configuration in Oracle Integration.

Oracle-BRT-ERP_KanbanBOX_PO_Config

This lookup table contains the information required to create a purchase order in Oracle Fusion Cloud Procurement. The following table provides the description for each parameter:

Name	Description
ApproverEmail	The email ID for the purchase order approval workflow as configured on Oracle Fusion Cloud Procurement. Example value: calvin.roth@oraclepdemos.com Update this value based on the configuration in Oracle Fusion Cloud Procurement.
Description	The description used in the mapping for the Requisition Header information. Example value: Requisition from KanbanBOX Update this value based on your requirements.
DocumentStatusCode	The standard value of the Document Status code while creating purchase requisitions as configured on Oracle Fusion Cloud Procurement. Example value: APPROVED
Note Do not modify this value.	
PreparerEmail	The email ID of the purchase-requisition preparer, which is to be used in the mapping. Example value: Nash.Lans@oraclepdemos.com Update this value based on the configuration in Oracle Fusion Cloud Procurement.

Name	Description
DestinationTypeCode	The standard value of the Destination Type code to be used in the mapping while creating purchase requisitions. Example value: INVENTORY  Note Do not modify this value.
KanbanAdmin	The Kanban Admin email ID that must be notified if there are errors while updating KanbanBOX. Example value: abc@xyz.com Update this value based on your KanbanBOX configuration.
PONotificationUserName	The user name of the procurement manager in Oracle Fusion Cloud Procurement who receives the Fusion pop-up notifications on purchase order creation; this value is case-sensitive. Example value: nash.lans Update this value based on the configuration in Oracle Fusion Cloud Procurement.
TaskCreator	The name of the procurement manager (task creator) displayed in the Fusion pop-up notifications for purchase orders. Example value: Nash Lans Update this value based on the configuration in Oracle Fusion Cloud Procurement.
OICAdmin	The Oracle Integration Admin email ID that must be notified on failures. Example value: abc@xyz.com Update this value based on the configuration in Oracle Integration.
BatchCardLimit	The threshold limit on the number of batch cards to be processed by Oracle Integration in a single Kanban release (via an integration call). Any incoming request having batch cards beyond this number is not processed by Oracle Integration. Example value: 5 Update this value based on your requirement.

Oracle-BRT-ERP_KanbanBOX_PO_MasterData

This lookup table contains the Requester information that's used for creating the purchase requisition on Oracle Fusion Cloud Procurement. Update the requester details according to your implementation.

Name	Description
RequesterEmail	The email ID of the Requester used to create the purchase requisition. Example value: kanban.user@oraclepdemos.com Update this value based on the configuration in Oracle Fusion Cloud Procurement.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the Configuration Editor, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Obtain the endpoint URLs for the trigger integration flows, namely **Oracle KanbanBOX ERP PurchaseOrder Create** and **Oracle KanbanBOX ERP WorkOrder Create**.
 - i. On the Configuration Editor page, select an integration flow and click **Run**.
 - ii. On the resulting pop-up dialog, click the **Metadata URL** link.
 - iii. From the new tab that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - b. Log in to your KanbanBOX instance, and configure it to perform the following actions when a Kanban card is released:
 - Send a POST request to the **Oracle KanbanBOX ERP PurchaseOrder Create** endpoint URL along with the purchase-requisition data. For example data, see [Example Request Payloads](#).
 - Send a POST request to the **Oracle KanbanBOX ERP WorkOrder Create** endpoint URL along with the work-order data. For example data, see [Example Request Payloads](#).

When the recipe receives these POST requests, it invokes the subsidiary flows in the package, creates a corresponding purchase order in Oracle Fusion Cloud Procurement and work order in Oracle Fusion Cloud Manufacturing, and finally sends the numbers of the orders created back to KanbanBOX.

- c. Log in to Oracle Fusion Cloud Procurement as the Procurement Manager, and verify the purchase requisitions and orders created. In addition, create a purchase-order receipt when a purchase order is fulfilled.
 - i. To view the purchase requisitions created, click the **Procurement** tab on the home page, and then click **Purchase Requisitions**.
 - ii. To view the purchase orders created, click the **Procurement** tab on the home page, and then click **Purchase Orders**. Note the order numbers generated for specific Kanban cards to later verify them.
 - iii. To view the purchase-order creation notifications, click the **Bell** icon in the top-right corner. The notifications display the purchase-order number for each KanbanBox card.

- iv. To create a purchase-order receipt when a purchase order is fulfilled:
 - On the home page, click the **Procurement** tab, and then click **My Receipts**.
 - On the Receive Items page, select the required purchase order line, and click **Receive**. (If required, search for a purchase order using its number).
 - On the Create Receipts page, enter the necessary data like quantity, waybill number, and so on, and then click Submit.

A receipt is created and its number is displayed. This action triggers the **Oracle ERP KanbanBOX PurchaseOrder Receipt** flow in Oracle Integration, which propagates the receipt status to KanbanBOX.

- d. Log in to Oracle Fusion Cloud Manufacturing as the Manufacturing Plant Manager, and verify the work orders created. In addition, mark a work order complete when it is fulfilled.
 - i. To view the work orders created:
 - Click the **Supply Chain Execution** tab on the home page, and then click **Work Execution**.
 - On the resulting page, click **Tasks** on the right pane, and then click **Manage Work Orders**.
 - On the Manage Work Orders page, search for the required work orders.
 - ii. To view the work-order creation notifications, click the **Bell** icon in the top-right corner. The notifications display the work-order number for each KanbanBox card.
 - iii. To mark a work order complete when it is fulfilled:
 - Click the **Supply Chain Execution** tab on the home page, and then click **Work Execution**.
 - On the resulting page, click **Tasks** on the right pane, and then click **Manage Work Orders**.
 - On the Manage Work Orders page, click **Tasks** on the right pane, and then click **Review Dispatch List**.
 - On the Review Dispatch List page, search for the required work order along with an associated work area (operation).
 - When the work-order row is displayed, expand it and click **Quick Complete**. Similarly, mark all the operations associated with the work order as complete.
 - Click **Done**.

The completion action triggers the **Oracle ERP KanbanBOX WorkOrder Status** flow in Oracle Integration, which propagates the work order status to KanbanBOX.

- 3. Monitor the running of the integration flow in Oracle Integration.
 - a. In the Configuration Editor, click **Activation** to view the integrations in the recipe. Hover over the integration to monitor, then click **Actions** **...** and select **Run**, then **Track instances**.
 - b. On the Track Instances page, observe the integration flow of the recipe being triggered and running successfully.
- 4. Finally, log in to your KanbanBOX instance, and check for the order numbers and statuses for each Kanban card.

On a Kanban card that has been released:

- a. View the purchase order number updated on the card. The number is updated in the field named, `order_number_for_customer`.
- b. View the status of the Kanban card. It's set to `Full` if the purchase order has been received.
- c. View the work order number updated on the card.
- d. View the work order completion status and quantity updated on the card.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Example Request Payloads

This section provides the example request payloads that must be sent from KanbanBOX to trigger the recipe. Using the payload information provided here, you can test the recipe for readiness in Oracle Integration.

Example: Purchase Requisition Payload

Here's an example payload to trigger the **Oracle KanbanBOX ERP PurchaseOrder Create** integration flow of the recipe.

```
{
  "kanbanbox_params": {
    "batch_cards": [{
      "part_unit_of_measurement": "Ea",
      "kanban_quantity": "1",
      "required_date": "2022-02-28",
      "customer_number": "K Store",
      "last_status_change_date": "2022-02-28",
      "type_of_kanban": "purchase",
      "business_unit": "US5 Business Unit",
      "requestor": "User, KanbanBox",
      "buyer": "Lans, Nash",
      "deliver_to_location": "Chicago 501",
      "kanban_status_name": "released",
      "organization": "501",
      "part_number": "G2A4009",
      "customer_name": "K Store",
      "supplier_name": "Allied Supplies",
      "ekanban_string": "R3GDA9C",
      "supplier_number": "300000230281607"
    }]
  }
}
```

Definition: Purchase Requisition Payload

The following table describes the attributes in the Purchase Requisition payload.

Attribute Name	Description
<code>part_unit_of_measurement</code>	The unit of measurement that is used to specify the part quantity in Oracle Fusion Cloud Procurement.

Attribute Name	Description
required_date	The required date for the part as entered in a purchase requisition. This field accepts the current date or a future date.
customer_number	This is mapped to the sub-inventory name in Oracle Fusion Cloud Procurement for the purpose of integration.
customer_name	This is mapped to the sub-inventory name in Oracle Fusion Cloud Procurement for the purpose of integration.
last_status_change_date	The field that provides information about the last-updated status change date. The field is not used or mapped in the recipe.
type_of_kanban	The attribute value purchase indicates that a purchase requisition needs to be created. The field is not used or mapped in the recipe.
business_unit	The business-unit name from where the purchase requisition is created in Oracle Fusion Cloud Procurement.
requestor	The user owning the requisition in Oracle Fusion Cloud Procurement. The field is not used or mapped in the recipe.
buyer	The person managing the purchase requisition in Oracle Fusion Cloud Procurement.
deliver_to_location	The location to deliver the goods in Oracle Fusion Cloud Procurement.
kanban_status_name	The status of the card from the Kanban system. For example, Released. The field is not used or mapped in the recipe.
organization	The code for the organization where goods are delivered.
part_number	The item number in Oracle Fusion Cloud Procurement.
supplier_name	The name of the supplier in Oracle Fusion Cloud Procurement. The field is not used or mapped in the recipe.
ekanban_string	The unique card-identifier generated and sent from KanbanBox.
supplier_number	The unique identifier representing the supplier in Oracle Fusion Cloud Procurement. The field is not used or mapped in the recipe.

Example: Work Order Payload

Here's an example payload to trigger the **Oracle KanbanBOX ERP WorkOrder Create** integration flow of the recipe.

```
{
  "kanbanbox_params": {
    "batch_cards": [{
      "part_unit_of_measurement": "Ea",
      "kanban_quantity": "40",
      "required_date": "2022-02-26",
      "customer_number": "K Store",
      "last_status_change_date": "2022-02-26",
      "type_of_kanban": "production",
      "kanban_status_name": "released",
      "organization": "501",
      "part_number": "G2A4008",
      "customer_name": "K Store",
      "supplier_name": "Allied Supplies",
      "ekanban_string": "PTA7RT59",
      "supplier_number": "300000230281607"
    }]
  }
}
```

Definition: Work Order Payload

The following table describes the attributes in the Work Order payload.

Attribute Name	Description
part_unit_of_measurement	The unit of measurement that is used to specify the part quantity in Oracle Fusion Cloud Manufacturing. The field is not used or mapped in the recipe.
kanban_quantity	The number of quantities placed to create the work order against the item pre-configured in Oracle Fusion Cloud Manufacturing.
required_date	The required date for the part as entered in a work order. This field accepts the current date or a future date.
last_status_change_date	The field that provides information about the last-updated status change date. The field is not used or mapped in the recipe.
type_of_kanban	The attribute value <code>production</code> indicates that a work order needs to be created.
kanban_status_name	The status of the card from the Kanban system. For example, <code>Released</code> . The field is not used or mapped in the recipe.
Organization	The code for the organization where goods are delivered.
part_number	The item number in Oracle Fusion Cloud Manufacturing.
customer_name	This is mapped to the sub-inventory name in Oracle Fusion Cloud Manufacturing for the purpose of integration. The field is not used or mapped in the recipe.
supplier_name	The name of the supplier in Oracle Fusion Cloud Manufacturing. The field is not used or mapped in the recipe.
ekanban_string	The unique card-identifier generated and sent from KanbanBOX.
supplier_number	The unique identifier representing the supplier in Oracle Fusion Cloud Manufacturing. The field is not used or mapped in the recipe.
customer_number	This is mapped to the sub-inventory name in Oracle Fusion Cloud Manufacturing for the purpose of integration.

Integrate Oracle Fusion Cloud IoT Intelligent Applications 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 Fusion Cloud IoT Intelligent Applications and Oracle Field Service Cloud.

Overview

This recipe enables you to dispatch technicians quickly and increase productivity by having activities automatically created, without any manual interventions, from Oracle Fusion Cloud IoT Intelligent Applications 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 Fusion Cloud IoT Intelligent Applications
- 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 Fusion Cloud IoT Intelligent Applications
- Oracle Integration

Install, Configure, and Run the Recipe

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

Configure resources of the recipe. See [Configure Resources in a Package](#).

Related Documentation

[Oracle Field Service Cloud / IoTCS Integration using OIC](#)

Integrate Oracle MICROS Reporting and Analytics Sales Data from ATP with Oracle Financials Accounting Hub

Use this recipe to create accounting journals in Oracle Financials Accounting Hub from Oracle MICROS Reporting and Analytics (R&A) sourced sales and operational data stored in Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe integrates Oracle Autonomous Transaction Processing (ATP) with Oracle Financials Accounting Hub.

This is a scheduled integration recipe that reads the Oracle MICROS Reporting and Analytics (R&A) data stored in Oracle ATP, and imports the data into Accounting Hub to create accounting journals.

For this recipe, you'll create sample tables in Oracle Autonomous Transaction Processing (ATP) and configure Oracle Financials Accounting Hub to receive and store data based on those tables. See Support note [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#) for the SQL scripts that you must run to create the tables. Note that the recipe can be extended to store and extract any available data points in ATP and Oracle Financials Accounting Hub, with the integration adapter configuration and mapping updated accordingly.

To use the recipe, you must install the recipe and configure the connections and other resources within it.

The recipe contains one integration flow:

- **Oracle ATP Sales to AH Journals:** This scheduled integration fetches data stored in ATP tables and generates Transaction header, line, and metadata files. Subsequently, it

compresses these files into zip archive and uploads them to Universal Content Management (UCM) using ERP Integration Service. It then places a request to submit the Oracle Enterprise Scheduler (ESS) job Import Accounting Transactions (XLATXNIMPORT), which imports the data into Oracle Financials Accounting Hub.

Assumptions

- You have installed and run the **Oracle MICROS R&A — Oracle ATP | Export Sales Data** recipe on your Oracle Integration instance.
- You have created all the ATP tables and views for the sales data by running the SQL scripts provided in Support note [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#).
- The table and view names are used as defined in [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#). If any change is made to the table or view names, you must modify the connector configuration in the integration accordingly.
- Recipe integration includes only sample error handling and notifications. Robust error handling based on customer needs must be implemented by the recipe consumer.
- Both recipes include a set of chosen dimensions, and sales and operational data points to enable the Oracle Financials integration. Recipe consumers can choose to extend the recipes by using additional data type invokes, which means that corresponding ATP tables and Accounting Hub configurations must be added/modified accordingly.

System and Access Requirements

- Oracle Integration, Version 22.1.3.0.0 or higher
- Oracle ERP Cloud, Version 21D or higher
- Oracle Autonomous Transaction Processing, Version 19c

Before You Install the Recipe

Perform the following configuration tasks to successfully install and run the recipe.

Configure Oracle ERP Cloud

To capture the sales data that must be accounted in Financials Cloud using Oracle Financials Accounting Hub:

1. In the Oracle General Ledger application:
 - Make sure you have the basic Enterprise Structure setup, which includes your Legal Entity, Primary Ledger, and Ledger options setup.
2. In the Oracle Financials Accounting Hub/Accounting Transformation application:
 - a. Create a subledger source system with short name FB_REV_ACCT with the following details.

Note

These objects must be created exactly as shown. Any changes made here must be incorporated in the integration configuration accordingly.

Object Column Name	Name	Level	Data Type
APPLICATION_ID	Application Identifier	Header	Number
BUS_DATE	Business Date Accounting Event	Header	Date
EVENT_ID	Identifier	Header	Number
LOCATION	Location	Header	Alphanumeric
LOC_CODE	Location Code	Header	Alphanumeric
REV_CENTER	Revenue Center	Header	Alphanumeric
REV_CENTER_CODE	Revenue Center Code	Header	Number
TRANSACTION_DATE	Transaction Date	Header	Date
TRANSACTION_NUMBER	Transaction Number	Header	Alphanumeric
CODE	Code	Line	Number
COUNT	Count	Line	Number
DEFAULT_AMOUNT	Default Amount	Line	Number
DEFAULT_CURRENCY	Default Currency	Line	Alphanumeric
DESCRIPTION	Code Description	Line	Alphanumeric
LINE_NUMBER	Line Number	Line	Number
MAJOR_GROUP	Major Group	Line	Alphanumeric
MAJOR_GROUP_CODE	Major Group Code	Line	Number
ORDER_TYPE	Order Type	Line	Alphanumeric
ORDER_TYPE_CODE	Order Type Code	Line	Number
PREP_COST	Preparation Cost	Line	Number

- b. Create mapping sets, account rules, description rules, supporting references, journal line rules, and journal line rule sets to create a subledger accounting method.

Note

Based on the Oracle Financials Accounting Hub configurations, the Accounting Hub File Based Data Import (FBDI) file configurations may have to be changed. Review the FBDI file configurations in Oracle Integration once the configurations are complete.

3. Create an integration user account on Oracle Financials Accounting Hub. To access Oracle SCM Cloud from Oracle Integration, you'll require a separate user account on Oracle SCM Cloud.

Log in to the Oracle SCM Cloud instance as an **Administrator** and perform the following tasks:

- Create a user account for Oracle Integration, which will be required to connect to SCM from Oracle Integration using the Oracle ERP Cloud Adapter.
- Assign roles to the user account. See *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
- In addition to the above roles, also assign the roles needed for Oracle Integration to perform the following operations in Oracle Financials Accounting Hub:
 - Upload file content to the UCM server

- Run the Import Accounting Transactions job

Configure Oracle Autonomous Transaction Processing

To access Oracle Autonomous Transaction Processing from Oracle Integration, perform the following configuration tasks on your ATP database:

1. Obtain the Cloud Wallet and the Credentials of the ATP instance to be used in the Oracle Integration connections.
2. Create a view with name FNB_AHCS_REPORT_FOR_CSV_V in your ATP database using FNB_AHCS_REPORT_FOR_CSV_V.sql and FNB_AHCS_LINES_V.sql attached to Support note [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud MICROS Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle SCM Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle SCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle ATP MICROS Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Service Name.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave JDBC Over SSL selected.
Wallet	Delete the existing wallet, then click Upload and browse for the cloud wallet zip file for your ATP connection. Click Upload .
Wallet Password	Specify the cloud wallet for your ATP instance.
Database Service Username	Enter the Database Service user name for your ATP instance.
Database Service Password	Enter the Database Service password for your ATP instance.

 Note

These configurations indicate a wallet-based connectivity to ATP. This may vary depending on the type of ATP and its authentication in your instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

Edit the lookup tables and enter necessary values.

This recipe contains the following lookup tables:

- **ORACLE-BRT-ATP_AH_SalesConfiguration**

This lookup table contains the information needed to import sales data from Oracle MICROS to Oracle Financials Accounting Hub.

Parameter Name	Description
ESSJobPackageName	The Oracle Enterprise Scheduler Job Package name of the job Import Accounting Transactions (XLATXNIMPORT) as configured on the Oracle Financials Accounting Hub instance. Example value: /oracle/apps/ess/financials/subledgerAccounting/shared/

 Note

Do not modify this value unless the Oracle Financials Accounting Hub configuration has changed.

Parameter Name	Description
ESSJobName	The Oracle Enterprise Scheduler Job Package name of the job Import Accounting Transactions (XLATXNIMPORT) as configured on the Oracle Financials Accounting Hub instance. Example value: XLATXNIMPORT Note Do not modify this value unless the Oracle Financials Accounting Hub configuration has changed.
LedgerName	The value of the Ledger Name where the sales transaction data is to be imported, as configured on Oracle Financials Accounting Hub. Example value: US Primary Ledger
TransactionsOutputDirectory	The Oracle Integration Stage Folder path where the Transaction template files are written and compressed before being uploaded to UCM. Example value: /FNB_ICS/TransactionFiles
TransactionsHeaderFileName	The name of the header file of the AH Transaction Upload Template. Example value: XlaTrxH.csv Note Do not modify this value unless the Oracle Financials Accounting Hub configuration has changed.
TransactionsLineFileName	The name of the line file of the AH Transaction Upload Template. Example value: XlaTrxL.csv Note Do not modify this value unless the Oracle Financials Accounting Hub configuration has changed.

Parameter Name	Description
TransactionsPropertiesFileName	The name of the metadata file of the AH Transaction Upload Template. Example value: XlaTrxL.csv

Note

Do not modify this value unless the Oracle Financials Accounting Hub configuration has changed.

- **ORACLE-BRT-Micros_ErrorNotification_Config**

This lookup table contains the information needed to trigger the error handlers.

Parameter Name	Description
OIC	The Oracle Integration Admin email address that must be notified on failures.
ERP	The ERP Admin email address that must be notified on failures.
Micros	The MICROS Admin email address that must be notified on failures.

1. In the Lookups section, click the lookup name.
2. Update parameter values as required.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle Financials Accounting Hub instance, and check if the sales data has been imported:

- a. Click **General Accounting** on the home screen. then click **journals**.
- b. Click **Tasks** to open the selection tray, then click **review Subledger Journals**.
- c. Enter the search parameters, such as Ledger, Journal Source, and Date and click **Search**.

Note

The values of these search parameters depend on the Oracle Financials Accounting Hub Journal configuration.

The transaction lines imported from Oracle MICROS Reporting and Analytics from ATP for the selected business date are displayed.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Integrate Oracle MICROS Reporting and Analytics with Oracle Autonomous Transaction Processing

Use this recipe to export restaurant sales and operational data from Oracle MICROS Reporting and Analytics to Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe integrates Oracle MICROS Reporting and Analytics with Oracle Autonomous Transaction Processing.

This is a scheduled integration recipe that extracts sales and dimensions data from Oracle MICROS Reporting and Analytics using its Business Intelligence APIs and imports the data into Oracle Autonomous Transaction Processing.

For the purpose of the recipe integration, sample tables are created in Oracle Autonomous Transaction Processing and the integration is configured to run SQL statements based on those tables. See Support note [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#) for the SQL scripts that you must run to create the tables. Note that the recipe can be extended to extract any available data points from Reporting and Analytics, with the integration adapter configuration and mapping updated accordingly.

To use the recipe, you must install the recipe and configure the connections and other resources within it.

The recipe contains three integration flows:

- **Oracle MICROS RnA to ATP Export Process:** This integration calls Oracle MICROS RnA Dimensions to ATP and Oracle MICROS RnA Sales to ATP to be executed consecutively. It acts as a scheduler integration.
- **Oracle MICROS RnA Dimensions to ATP:** This integration fetches all the location information from Reporting and Analytics and inserts it into a dimensions table in ATP. Then, for each location, it inserts all the location dimension details (Tender Media Dimensions, Service Charge Dimensions, Order Type Dimensions, Discount Dimensions, Revenue Center Dimensions, Menu Item Dimensions, and Tax Dimensions) into the corresponding tables in ATP. For the list of tables, see [Configure Oracle Autonomous Transaction Processing](#).
- **Oracle MICROS RnA Sales to ATP:** This integration fetches all the location dimensions from Reporting and Analytics. Then, for each location, it gets the latest business date and for that particular date, inserts sales and operational data (Menu Item Daily Totals, Service Charge Daily Totals, Tax Daily Totals, Tender Media Daily Totals, and Discount Daily Totals) into the corresponding tables in ATP. For the list of tables, see [Configure Oracle Autonomous Transaction Processing](#).

Assumptions

- You have created all the tables for the dimensions, and sales and operational data, as described in Support note [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#).
- The table and view names are used as defined in [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#). If any change is made to the table or view names, you must modify the connector configuration in the integration accordingly.
- Recipe integration includes only sample error handling and notifications. Robust error handling based on customer needs must be implemented by the recipe consumer.
- The recipe include a set of chosen dimensions, and sales and operational data points to enable the Oracle Financials integration. Recipe consumers can choose to extend the recipe by using additional API invokes, which means that corresponding ATP tables must be added/modified accordingly.

System and Access Requirements

- Oracle Integration, Version 22.1.3.0.0 or higher
 - An account in Oracle Integration with the Service Invoker role to configure the recipe
- Oracle Autonomous Transaction Processing, Version 19c
- Oracle MICROS Reporting and Analytics, Version 20.1 or higher
 - Oracle MICROS Symphony Cloud Service, Business Intelligence Interface (BI API)

Before You Install the Recipe

Perform the following configuration tasks to successfully install and run the recipe.

Configure Oracle MICROS Reporting and Analytics

1. Configure a Business Intelligence API account in Reporting and Analytics to be used in the Oracle Integration connections, as described in [Business Intelligence API](#) in the *Reporting and Analytics User Guide*.
2. Configure the Reporting and Analytics OAuth server to send a callback response to the Oracle Integration instance where this recipe is being installed.
 - This requires the Oracle consultant to open an internal ticket and request configuring multiple redirect URLs.
 - The following information needs to be provided:
 - a. Customer Environment Name
 - b. Enterprise Name and Enterprise Short Name
 - c. Client_ID of the BI API Account
 - d. Custom redirect URL to be configured. For example: `https://oic-dev-abcdefghi1234.integration.us-phoenix-1.ocp.oraclecloud.com/icsapis/agent/oauth/callback`
3. Collect the following information from Reporting and Analytics to configure the Oracle Integration connections. For details, see [Send Requests](#) in the *Business Intelligence API Guide*.
 - Account name
 - Password
 - Enterprise short name (orgIdentifier)
 - client_id
 - rest_server URLs for the Authentication APIs and the Business Intelligence data APIs

Besides the above, generate a code verifier and code challenge.

- The code_challenge is a Base64-encoded SHA-256 hash of the code_verifier. The code_verifier is a cryptographically random string.
- Both code_verifier and the code_challenge are generated by the client (in this case by the OIC recipe).
- The code_challenge and code_verifier are used to perform a challenge-response validation in OAuth PKCE Flow.
- code_challenge_method is fixed S256

For more information, see [Authenticate](#) in the *Business Intelligence API Guide*.

Configure Oracle Autonomous Transaction Processing

To access ATP from Oracle Integration, perform the following configuration tasks on your ATP database:

1. Obtain the Cloud Wallet and the Credentials of the Oracle Autonomous Transaction Processing instance to be used in the Oracle Integration connections
2. Create the following tables, which are required for collecting dimension and sales data. The DDL statements to create these tables are provided in Support note [Supplemental Code for Oracle MICROS – ATP – Accounting Hub Oracle Integration Cloud Recipes](#):

- FNB_DIM_LOCATION
- FNB_DIM_DISCOUNT
- FNB_DIM_MENUITEM
- FNB_DIM_ORDERTYPE
- FNB_DIM_REVENUECENTER
- FNB_DIM_SERVICECHARGE
- FNB_DIM_TAX
- FNB_DIM_TENDERMEDIA
- FNB_MENUITEMDAILYTOTALS
- FNB_SERVICECHARGEDAILYTOTALS
- FNB_TAXDAILYTOTALS
- FNB_TENDERMEDIADAILYTOTALS
- FNB_DISCOUNTDAILYTOTALS

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST OIC API Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the host name of the Oracle Integration instance where you installed the recipe.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
User Name	Enter the user name of the Oracle Integration user account with the Service Invoker role. See System and Access Requirements .
Password	Enter the password of the Oracle Integration user account with the Service Invoker role.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle MICROS Reporting and Analytics BI API Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the domain used to log into the Reporting and Analytics instance. For more information, see Send Requests in the <i>Business Intelligence API Guide</i> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Custom Three Legged Flow .
Authorization Request	Enter the authorization request format. Example: <code>https://BI_API_IDM_Server_HOST/oidc-provider/v1/oauth2/authorize?response_type=code&client_id= client_id &scope=openid&redirect_uri=https://OIC_HOST/icsapis/agent/oauth/callback&code_challenge=code_challenge&code_challenge_method=S256</code>
Access Token Request	Enter the access token request format. Example: <code>-X POST -H 'Content-Type: application/x-www-form-urlencoded' -d 'grant_type=authorization_code&code=\${auth_code}&redirect_uri=https://OIC_HOST/icsapis/agent/oauth/callback&code_verifier=code_verifier&client_id=client_id' https://BI_API_IDM_Server_HOST/oidc-provider/v1/oauth2/token</code>
Refresh Token Request	Enter the refresh token request format. Example: <code>-X POST -H 'Content-Type: application/x-www-form-urlencoded' -d 'grant_type=refresh_token&refresh_token=\${refresh_token}&redirect_uri=https://OIC_HOST/icsapis/agent/oauth/callback&code_verifier=code_verifier&client_id=client_id' https://BI_API_IDM_Server_HOST/oidc-provider/v1/oauth2/token</code>
\$auth_code	code
\$access_token	access_token
\$refresh_token	refresh_token
\$expiry	expires_in
\$token_type	token_type
access_token_usage	<code>-H Authorization: \${token_type} \$ {access_token}</code>

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Service Name.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Upload the wallet zip file.
Wallet Password	Enter the password for the wallet.
Database Service Username	Enter the service user name of the Database created for Oracle Integration on Oracle Autonomous Transaction Processing.
Database Service Password	Enter the password of the Database created for Oracle Integration on Oracle Autonomous Transaction Processing.

Note

These configurations indicate a wallet-based connectivity to ATP. This may vary depending on the type of ATP and its authentication mechanism used in your instance. Configure the connection accordingly.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORACLE-BRT-Micros_ErrorNotification_Config** lookup table and enter the necessary values.

1. In the Lookups section, click the lookup name.
2. The **ORACLE-BRT-Micros_ErrorNotification_Config** lookup table contains the information needed to trigger the error handlers. Enter the required information for the following parameters:

Parameter	Information to Enter
OIC	The Oracle Integration Admin email address that must be notified on failures.
ERP	The Oracle Integration Admin email address that must be notified on failures.
Micros	The Oracle Integration Admin email address that must be notified on failures.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle MICROS RnA to ATP Export Process** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Log in to your ATP database:
 - a. Navigate to the ATP schema.
 - b. Under the corresponding tables, notice that the dimensions and sales data for the latest business date is inserted from Reporting and Analytics to ATP.
4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)
- [Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3](#)
- [Business Intelligence API Guide](#)
- [Reporting and Analytics User Guide](#)

Integrate with DocuSign

Use these recipes to create a draft envelope in your DocuSign account and retrieve the status of an existing DocuSign envelope from an external application.

These recipes are available in Oracle Integration as:

- **REST — DocuSign | Create Envelope**

- **REST — DocuSign | Get Envelope Status**

Topics:

- [About the Recipes](#)
- [Before You Install the Recipes](#)
- [Install and Configure the Recipes](#)
- [Activate and Run the Recipes](#)

About the Recipes

With these recipes, you can:

- Create a draft envelope containing a document (for example, a PDF) 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 use the recipes, you must install the recipes and configure the connections within them. Subsequently, you can activate the recipes and trigger their respective integration flows through REST requests with necessary request parameters.

Integration	Description	Parameters required to run the recipe
Oracle REST DocuSign Create Envelope	Creates a draft envelope containing a document in your DocuSign account.	Provide the following parameters in the request message: • document id: Specify the ID of the document you want to insert within a DocuSign envelope • name: Specify the name of the document. • documentBase64: Specify the Base64 value for the document.
Oracle REST DocuSign Get Envelope Status	Retrieves the status of an existing envelope in your DocuSign account.	Provide the ID of the DocuSign envelope of which you want the status.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- DocuSign
- An account on DocuSign with the Administrator role

Before You Install the Recipes

You must perform the following configuration tasks on your DocuSign instance in order to connect to it using Oracle Integration and successfully create a draft envelope or retrieve the status of an existing DocuSign envelope.

Configure DocuSign

To connect to DocuSign from Oracle Integration, 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.

7. 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 Recipes

On your Oracle Integration instance, install the recipes to deploy and configure the integrations and associated resources.

Install the recipes on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipes, configure the following resources:

Configure the DocuSign Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the 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 .

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.
You'll now be redirected to the DocuSign login page.
6. Enter your DocuSign account credentials and click **Continue** to log in.
You're prompted to authorize the connection to the DocuSign application created previously.
7. 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.
8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
9. To return to the project workspace, click **Go back** .

Configure the REST Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipes

After you've configured the connections and other resources, you can activate and run the recipes.

1. Activate the recipes. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send the required REST command to this endpoint URL along with the request parameters. See **Parameters required to run the recipe** in [About the Recipes](#).
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, enter the parameters required to run the recipe.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success status 200 OK.
4. View the response payload in the Activity stream to verify if the action is successful.
 - For the **REST — DocuSign | Create Envelope** recipe, check if the response returned contains the new envelope ID.
 - For the **REST — DocuSign | Get Envelope Status** recipe, check if the response returned contains the status and the last status-change date and time.

Related Documentation

- *Using the DocuSign Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Interact with LinkedIn

Use these recipes to get user profile and share content on LinkedIn, according to REST requests received from an external application.

These recipes are available in Oracle Integration as:

- **REST — LinkedIn | Get User Lite Profile**
- **REST — LinkedIn | Post Content**
- **REST — LinkedIn | Post a Video**

Topics:

- [About The Recipes](#)
- [Before You Install the Recipes](#)
- [Install and Configure the Recipes](#)
- [Activate and Run the Recipes](#)

About The Recipes

The LinkedIn recipes use the REST adapter and the LinkedIn adapter to create an integration with the LinkedIn application. The recipes enable you to use the basic LinkedIn functions such as getting user's lite profile and posting content. You can perform these operations on behalf of an authenticated user.

Field	Description	Parameters Required to Run the Recipe
Get user lite profile	Retrieves the lite profile of the current LinkedIn user. Scope: r_liteprofile	No parameter is required in the request message to run this recipe.

Field	Description	Parameters Required to Run the Recipe
Post content	Uses the LinkedIn Shares API to post content from the current LinkedIn user's account.	Not e Linke dIn still supp orts the Shar es API, but plans to depr ecate it in the near futur e. Also, the Shar e API does n't supp ort video . <pre>{ "content": { "title": "Test Oracle LinkedIn Share message" }, "distribution": { "linkedInDistributionTar get": {} }, "owner": "<your_linkedIn_account_ urn>", "subject": "Test Oracle LinkedIn - Share 005", "text": { "text": "Test Oracle LinkedIn Share message" } }</pre> Provide the following sample payload in the request message:
Scope: r_emailaddress		

Field	Description	Parameters Required to Run the Recipe
Post a video	Uses the LinkedIn User Generated Content (UGC) Posts API to post a video from the current LinkedIn user's account.  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.	Provide the following sample payload in the request message: <pre>{ "author": "<your_LinkedIn_account_urn>", "lifecycleState": "PUBLISHED", "specificContent": { "com.linkedin.ugc.ShareContent": { "shareCommentary": { "text": "Test Oracle LinkedIn Share User Content" }, "shareMediaCategory": "NONE" }, "visibility": { "com.linkedin.ugc.Member NetworkVisibility": "PUBLIC" } } }</pre>

Scope: w-member-social

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0, or higher
- LinkedIn
- An account on LinkedIn with the Administrator role

Before You Install the Recipes

Create a LinkedIn application to get access to the LinkedIn API, which is then used to configure the LinkedIn connection. See *Prerequisites for Creating a Connection in Using the LinkedIn Adapter with Oracle Integration 3*.

Install and Configure the Recipes

On your Oracle Integration instance, install the recipes to deploy and configure the integrations and associated resources.

Install the recipes on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipes, configure the following resources:

Configure the REST LinkedIn Connection

1. In the Connections section, click the connection name.
2. In the Security section, leave **OAuth 2.0 or Basic Authentication** selected.
3. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Configure the LinkedIn Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring LinkedIn. See Before You Install the Recipes .
Client Secret	Enter the client secret obtained while configuring LinkedIn.
Scope	Enter the scope of the access request. See the table in About The Recipes for the scopes of different recipes.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.
You'll now be redirected to the LinkedIn sign in page.
6. Enter your LinkedIn account credentials.
A page is displayed asking for permission to interact with the account.
7. Click **Yes**.
You're informed that access is allowed. You can now switch back to the LinkedIn Connection window of Oracle Integration to test your connection.

8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
9. To return to the project workspace, click **Go back** .

Activate and Run the Recipes

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipes. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send the required REST command to this endpoint URL along with the request parameters. See **Parameters required to run the recipe** in [About The Recipes](#).
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, enter the parameters required to run the recipe.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success response, 200 OK.
4. Verify if the action is successful in your LinkedIn instance. For example, if you had sent a POST request with the desired text to be posted, check if the content is posted on the authenticated user's LinkedIn instance.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the LinkedIn Adapter with Oracle Integration 3*

Interact with Slack

Use these recipes to create a channel and post a message to a channel on Slack, according to REST requests received from an external application.

These recipes are available in Oracle Integration as:

- **REST — Slack | Create a Channel**
- **REST — Slack | Post a Message**

Topics:

- [About the Recipes](#)
- [Before You Install the Recipes](#)
- [Install and Configure the Recipes](#)
- [Activate and Run the Recipes](#)

About the Recipes

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 connections 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 a 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 a Message** recipe, you must supply the message 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.

Recipe Name	Description	REST Commands and Example Payloads
Create a Channel	Creates a public or private channel for communication on the Slack application.	REST API Command: POST: https://<host:port>/ic/api/integration/v2/flows/rest/project/ORCL-R-SLACK_CREATE_CHANNEL/ORCL-R-SLACK_CREATE_CHANNEL/1.0/createChannel Example Request Payload: <pre>{ "channelName": "New-Channel", "is-private": "true" }</pre> Example Response Payload: <pre>{ "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", "name_normalized": "new-channel", "is_shared": "false", "creator": "U01PUKGN6SK", "is_ext_shared": "false", "is_org_shared": "false", "is_pending_ext_shared": "false", "is_member": "true", "is_private":</pre>

Recipe Name	Description	REST Commands and Example Payloads
		<pre>"true", "is_mpim": "false", "last_read": "0000000000.000000" } }</pre>
Post a Message	Posts a message to a public or private Slack channel.	REST API Command: POST: https://<host:port>/ic/api/integration/v2/flows/rest/project/ORCL-R-SLACK_PUSH_NOTIFICATION/ORCL-R-SLACK_PUSH_NOTIFICATION/1.0/postMessage Example Request Payload: <pre>{ "Message": "Test Message", "channelName": "new- channel" }</pre> Example Response Payload: <pre>{ "ok": "true", "channel": "C01SGRKT84E", "ts": "1617031570.000300", "message": { "bot_id": "B01SC4EJ4JU", "type": "message", "text": "Test Message", "user": "", "ts": "1617031570.000300", "team": "" } }</pre>

System and Access Requirements

- Oracle Integration, Version 23.6.0.0.0, or higher

- Slack
- An account on Slack with the Administrator role

Before You Install the Recipes

To access Slack using Oracle Integration, you must perform certain configuration tasks in your Slack instance.

1. Create a Slack workspace.
 - a. Open the Slack get started page using the following URL: <https://slack.com/get-started#/createnew>
 - b. Enter your email and click **Continue**.
 - c. Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.
2. Create a Slack App.
 - a. Log in to the Slack app platform at <https://api.slack.com>.
 - b. In the Slack API title bar, click **Your Apps**.
 - c. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
 - d. In the Name app & choose workspace dialog, enter the following details:
 - i. In the **App Name** field, enter the name of your app.
For example, Oracle Integration App.
 - ii. In the **Pick a workspace to develop your app in** field, select your workspace from the drop-down list.
 - e. Click **Create App**.
The app gets created and its Basic Information page appears.
3. Get the Slack App API credentials.
 - a. In the app's Basic Information page, scroll down to the App Credentials section.
 - b. Note the client ID in the **Client ID** field, and the client secret in the **Client Secret** field.
You'll use these later while configuring connections to your Slack instance from Oracle Integration.
4. 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:
 - i. channels:write
 - ii. groups:write

iii. im:write

iv. mpim: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 Recipes

On your Oracle Integration instance, install the recipes to deploy and configure the integrations and associated resources.

Install the recipes on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipes, configure the following resources:

Configure the Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Before You Install the Recipes .
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	In the Scope field, enter channels:write groups:write im:write mpim:write.

3. Click **Provide Consent**.
A new browser window opens to approve access to Slack.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipes

After you've configured the connections and other resources, you can activate and run the recipes.

1. Activate the recipes. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send the required REST command to this endpoint URL along with the request parameters. See *REST Commands and Example Payloads* in [About the Recipes](#).
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, enter the parameters required to run the recipe.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success response, 200 OK.
4. Verify if the action is successful in your Slack instance. For example, if you had sent a POST request to create a Slack channel, check if the channel is created in your Slack instance.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

Interact with Twitter

Use these recipes to post a tweet and perform other actions on Twitter, according to REST requests received from an external application.

These recipes are available in Oracle Integration as:

- **REST — Twitter | Post a Tweet**
- **REST — Twitter | Retweet a Post**
- **REST — Twitter | Look Up Tweet Status**
- **REST — Twitter | Get Retweeted Messages**
- **REST — Twitter | Get User IDs That Retweeted a Tweet**
- **REST — Twitter | Find Follower IDs**

- **REST — Twitter | Find Friend IDs**
- **REST — Twitter | Get Locations with Trending Topics**
- **REST — Twitter | Get Trends for a Location**

Topics:

- [About The Recipes](#)
- [Before You Install the Recipes](#)
- [Install and Configure the Recipes](#)
- [Activate and Run the Recipes](#)

About The Recipes

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.

Field	Description	Parameters Required to Run the Recipe
Post a Tweet	Posts a tweet on your Twitter instance.	Provide the status parameter with the desired text to be posted in the tweet.
Retweet a Post	Reposts or forwards a tweet that was previously posted on Twitter.	Provide the ID parameter with the unique ID of the tweet to be retweeted.
Look Up Tweet Status	Returns the details of up to 100 tweets per request, as specified by the comma-separated values passed to the ID parameter. This recipe is useful to get the details of a collection of tweet IDs.	Provide the following parameters in the request message: • ID: Specify the tweet ID. • trim_user: When set to true (true/t/1), then it includes only the tweet author's 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).
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 User IDs That Retweeted a Tweet	Returns a collection of up to 100 user IDs belonging to users who have retweeted the tweet specified in the ID parameter.	Provide the ID of the tweet for which you wish to retrieve the list of accounts that retweeted it.

Field	Description	Parameters Required to Run the Recipe
Get Follower IDs	Returns the details of each user who is following an account.	Provide the <code>user_id</code> and the <code>screen_name</code> of the Twitter account in the request message. Note that the <code>screen_name</code> is the Twitter ID without the <code>@</code> sign.
Get Friend IDs	Returns the details of each user whom the current account is following (otherwise known as their "friends").	Provide the <code>user_id</code> and the <code>screen_name</code> of the Twitter account in the request message. Note that the <code>screen_name</code> is the Twitter ID without the <code>@</code> sign.
Get Trends for a Location	Returns the top 50 trending topics for the location specified by the Where On Earth Identifier (WOEID), if trending information is available for it.	Provide the ID parameter with the WOEID in the request message. 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 Locations with Trending Topics	Returns the locations that Twitter has trending topic information for. 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.	No parameter is required in the request message to run this recipe.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0, or higher
- Twitter
- An account on Twitter with the Administrator role

Before You Install the Recipes

Create a Twitter application to get access to the Twitter API, which is then used to configure the Twitter Adapter. See *Prerequisites for Creating a Connection in Using the Twitter Adapter with Oracle Integration*.

Install and Configure the Recipes

On your Oracle Integration instance, install the recipes to deploy and configure the integrations and associated resources.

Install the recipes on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipes, configure the following resources:

Configure the REST Twitter Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the connection URL in the following format: https:// <oracle_integration_server_name>:<port_number>

3. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Configure the Twitter Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Consumer Key (API Key)	Enter the API key obtained while configuring Twitter. See Before You Install the Recipes .
Consumer Secret (API Secret)	Enter the API secret obtained while configuring Twitter.
Access Token	Enter the access token obtained while configuring Twitter.
Access Secret	Enter the access secret obtained while configuring Twitter.

3. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipes

After you've configured the connections and other resources, you can activate and run the recipes.

1. Activate the recipes. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send the required REST command to this endpoint URL along with the request parameters. See **Parameters Required to Run the Recipe** in [About The Recipes](#).

3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, enter the parameters required to run the recipe.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success response, 200 OK.
4. Verify if the action is successful in your Twitter instance. For example, if you had sent a POST request with the desired text to be posted as tweet, check if the tweet has been posted on your Twitter account.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Twitter Adapter with Oracle Integration 3*

Invoke a Local Integration

Use this recipe to learn how you can utilize an integration flow to invoke another integration flow within a recipe.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how you can use one integration flow to invoke another local integration flow. The recipe contains two integration flows: **Oracle Hello World Map Data Local Invoke** and **Oracle REST Hello World Sample**.

To use the recipe, you must install the recipe and configure the connections within it. Subsequently, you can activate the first integration flow **Oracle Hello World Map Data Local Invoke**, and send a GET request (containing a name and email address) to the integration flow's endpoint URL from an external application. When triggered, the first integration flow invokes the second integration flow **Oracle REST Hello World Sample**, and passes the payload details to it. In turn, the second integration flow sends a sample message to the email address specified in the request parameter. A response is returned to the external application.

System and Access Requirements

- Oracle Integration, Version 24.02 or higher

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Sample Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the REST Sample Endpoint Interface Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a GET request to this endpoint URL along with the following request parameters:
 - i. **name**: Specify the name of the user.
 - ii. **email**: Specify the email address to which the message must be sent.
3. Test the recipe in Oracle Integration.

- a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the name of the user and specify the email address to which the message must be sent.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success response, 200 OK, and the response returned.
4. Check if a message is received on the email address specified in the request payload.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)

Invoke a Non-Local Integration

Use this recipe to learn how you can utilize an integration flow to invoke another integration flow within a different recipe.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how you can use one integration flow to invoke another non-local integration flow. For this purpose, the recipe's integration flow invokes the integration flow of the **REST — Oracle Integration | Send a Hello Message** recipe.

The recipe contains an application-driven integration flow, which uses the standard REST Adapter.

To use the recipe, you must install the recipe and configure the connections within it. Subsequently, you can activate the integration flow of the recipe and send a GET request (containing a name and email address) to the integration flow's endpoint URL from an external application. When triggered, the integration flow invokes a child flow (through another REST Adapter) and passes on the payload details. In turn, the child integration flow sends a sample message to the email address specified in the request parameter. A response is returned to the external application.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0, or higher

Before You Install the Recipe

In order to run this recipe, you must install and activate the **REST — Oracle Integration | Send a Hello Message** recipe on your Oracle Integration instance.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Sample Hello World Invoke Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the base URL of your Oracle Integration instance in the following format: <code>https://<OICHost>:<port></code> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Basic Authentication selected.
Username	Enter the user name of your Oracle Integration account.
Password	Enter the password of your Oracle Integration account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured.

A message confirms if your test is successful.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a GET request to this endpoint URL along with the following request parameters:
 - i. **Name**: Specify the name of the user.
 - ii. **Email**: Specify the email address to which the message must be sent.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the name of the user and specify the email address to which the message must be sent.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success response, 200 OK, and the response returned.
4. Check if a message is received on the email address specified in the request payload.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)

Invoke Pop-Up Notifications for Events in Oracle Fusion Applications

Use this recipe to invoke notifications for user-defined events in Oracle Fusion Applications. This recipe works with all Oracle Fusion Applications, such as SCM, ERP, HCM, and CX.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe enables you to invoke notifications in your Oracle Fusion Application for user-defined events. You can send notifications to specific Fusion Application users after certain events, such as creation of Fusion records (like sales orders) using APIs, either through Oracle Integration or external applications.

This is an application-driven recipe that creates a new notification in an Oracle Fusion Application when it receives a REST request with the notification data from an external application or from another integration flow in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate the integration flow of the recipe and send a REST request with the notification data to the integration's endpoint URL from an external application or from another integration flow in Oracle Integration. When triggered, the recipe's integration flow creates a notification in your Oracle Fusion Application using the data received and returns a success or failure response.

The notification shows up on the Fusion Application's home page for the specified user. You can customize the notification's message as required by any scenario.

System and Access Requirements

- Oracle Integration, Version 22.1.3.0.0 or higher
- An account on Oracle Integration with the Service Invoker role to configure the recipe
- Oracle Fusion Applications, Version 21D or higher
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

To access an Oracle Fusion Application from Oracle Integration and invoke notifications for events, you'll require a separate user account on the Oracle Fusion Application.

Log in to the Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. See Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
3. Additionally, assign the Human Capital Management Integration Specialist role to the account to provide access to the User Accounts APIs. See the support document: **2610503.1**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST OIC API Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the host name of the Oracle Integration instance in which you've installed the recipe.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
User Name	Enter the user name of the Oracle Integration user account with the Service Invoker role. See Before You Install the Recipe .
Password	Enter the password of the Oracle Integration user account with the Service Invoker role.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle ERP Cloud Notification Connection

1. In the Connections section, click the connection name.
2. In the Connection Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle ERP Cloud. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the notification data. Provide the notification data in the POST request's **Body** field. See the subsequent step for an example request payload.

The recipe invokes a notification in your Oracle Fusion Application for the user account specified in the notification data.

3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** on the integration flow, then select **Run**.

The Configure and run page opens.
 - b. In the Request section, click **Body** and enter the notification data. The data must contain the following elements:
 - **UserName**: Enter an existing user in your Oracle Fusion Application for whom the notification is intended. This field is case-sensitive.
 - **Message**: Enter a message to show in the notification.
 - **TaskCreator**: Enter the name of the notifier.

Here's an example data set:

```
{
  "UserName": "Robert.James",
  "Message": "Testing the Notification Recipe",
  "TaskCreator": "Admin User"
}
```

- c. Click **Test**.

The recipe invokes a notification in your Oracle Fusion Application for the user account specified in the notification data. The Activity Stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section, under the **Body** tab, view the success response, 200 OK.
4. Log in to your Oracle Fusion Application instance with the user account specified in the Request payload previously, and check for the new notification.
 - a. On the home page, the **Bell** icon in the top right corner displays a new notification.

Click the icon to view the notification's content. The notification also contains the name of the notifier, specified in the Request payload.

- b. You can also view the same notification under the Things to Finish section on the home page.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Log a Request Message

Use this recipe to log an external request message in Oracle Integration, using a REST Adapter trigger.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how you can log a request message from an external application in Oracle Integration, using the REST adapter as a trigger.

This is an application-driven recipe that creates a log message in Oracle Integration when it receives a REST request with the message data from an external application.

To use the recipe, you must install the recipe and configure the connections within the recipe. Subsequently, you can activate the integration flow of the recipe and send a REST request with the message data to the integration's endpoint URL from an external application. When triggered, the recipe's integration flow returns a response to the external application and also logs the request message in Oracle Integration, which you can view in the activity stream.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0, or higher

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a POST request to this endpoint URL along with the request message data. Provide the message data in the POST request's **URI parameters** section.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the request message.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, under the **Body** tab, you'll find a success response, 200 OK.
4. View the request message in the activity stream. In the Activity stream pane, click **View payload**  on the **Logger logMessage** item to view the request message.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)

Manage Objects in OCI Object Storage

Use this recipe to manage objects in Oracle Cloud Infrastructure Object Storage with a native action.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe can be used to upload, download, delete or list objects in OCI Object Storage.

To use the recipe, you must install the recipe and configure the resources within the recipe. The recipe uses a native action **OCI Object Storage** to connect to OCI Object Storage service. You don't need to configure an explicit REST adapter connection. The following operations can be used to manage objects in OCI Object Storage:

- **Upload** a file into OCI Object Storage
- **Download** a file from OCI Object Storage
- **List** files in OCI Object Storage
- **Delete** a file in OCI Object Storage

System and Access Requirements

- Oracle Integration 3
- OCI Object Storage
- Access to an OCI Object Storage bucket

Before You Install the Recipe

Create a dynamic group and policies to use the OCI Object Storage native action.

See **Prerequisites** in Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources.

Configure the Lookup Table

Edit the lookup table and enter appropriate values for the lookup keys.

1. In the Lookups section, click the lookup name.
2. Enter appropriate values for the look up keys.

Key	Value
BucketName	Specify the name of the OCI Object Storage bucket where you want to manage objects.
EmailFrom	Enter the email address from which error notifications are to be sent.
EmailTo	Enter the email address to which error notifications are to be sent.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** . . . on the **Oracle Object Storage Handler** integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, select **File** and then upload the object to add to OCI Object Storage (for example, a jpg file).
 - c. Click **Run**
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if the object (jpg file) has been added to the specified bucket in OCI Object Storage.
 - a. Go to Oracle Cloud Console.
 - b. In the navigation pane, select **Storage** and then **Buckets**.
 - c. Select the bucket and check if the file that you uploaded is displayed under the Objects section.

Related Documentation

- [OCI Object Storage documentation](#)

Manage ServiceNow Incidents

Use this recipe to create, update, or delete incidents in ServiceNow.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe can be used to create an incident, update an incident, or delete an incident in ServiceNow.

To use the recipe, you must install the recipe and configure the connections within it. The recipe contains three integrations: **Oracle REST ServiceNow Create Incident**, **Oracle REST ServiceNow Update Incident**, and **Oracle REST ServiceNow Delete Incident**.

- When the **Oracle REST ServiceNow Create Incident** integration flow is triggered by a REST request with `incidentdescription` as a request parameter, the recipe creates an incident in ServiceNow. The response containing the incident number and the incident Id is returned to you.
- When the **Oracle REST ServiceNow Update Incident** integration flow is triggered by a REST request with `incidentId`, `incidentState`, and `incidentUrgency` as request parameters, the recipe updates the corresponding incident details in ServiceNow.
- When the **Oracle REST ServiceNow Delete Incident** integration flow is triggered by a REST request with `incidentId` as a request parameter, the recipe deletes an incident in ServiceNow.

Integration	Description	REST Commands and Example Payloads
Oracle REST ServiceNow Create Incident	Creates an incident in the ServiceNow instance.	REST API Command: POST: https:// <host:port>/ic/api/ integration/v2/flows/ rest/project/ORCL-R- REST_SNOW_MANAGE_INCIDEN T/ORCL-R- REST_SERVIC_UPDATE_INCID E/1.0/create Example Request Payload: <pre>{ "incidentdescription" : "Employee phone number missing" }</pre> Example Response Payload: <pre>{ "serviceNowNr": "INC0010031", "serviceNowSysId": "fe6a74e62f503010f129811 df699b6e7" }</pre>

Integration	Description	REST Commands and Example Payloads
Oracle REST ServiceNow Update Incident	Updates an incident in the ServiceNow instance.	REST API Command: POST: https:// <host:port>/ic/api/ integration/v2/flows/ rest/project/ORCL-R- REST_SNOW_MANAGE_INCIDEN T/ORCL-R- REST_SERVIC_UPDATE_INCID E/1.0/update Example Request Payload: <pre>{ "incidentId": "bec4ce452f603010f129811df699b6cc", "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 = Resolved • 7 = Closed • 8 = Canceled incidentUrgency • 1 = High • 2 = Medium • 3 = Low Example Response Payload: <pre>{ "incidentStatus": "bec4ce452f603010f129811df699b6cc" }</pre>

Integration	Description	REST Commands and Example Payloads
Oracle REST ServiceNow Delete Incident	Deletes an incident in the ServiceNow instance.	REST API Command: POST: https:// <host:port>/ic/api/ integration/v2/flows/ rest/project/ORCL-R- REST_SNOW_MANAGE_INCIDEN T/ORCL-R- REST_SERVIC_UPDATE_INCID E/1.0/delete Example Request Payload: <pre>{"incidentId": "fe6a74e62 f503010f129811df699b6e7" }</pre> Example Response Payload: <pre>{ "status": "1" }</pre>

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- ServiceNow
- An account on ServiceNow with the Administrator role

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ServiceNow Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the ServiceNow instance URL in the **ServiceNow Instance Name** field. For example: https://host_name.service-now.com
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	By default, Basic Authentication is selected.
Username	Enter the user name of your ServiceNow instance.
Password	Enter the password of your ServiceNow instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send the required REST command to this endpoint URL along with the request parameters. See **REST Commands and Example Payloads** in [About This Recipe](#).
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under **URI parameters**, enter the request parameters.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.

- d. In the Response section of the Configure and run page, you'll find a success response 200 OK.
4. View the response payload in the Activity stream to verify if the action is successful. For example, if you had sent a POST request with the required request parameters to create an incident in ServiceNow, check if the incident number and incident Id is returned in the response.

Related Documentation

- *Using the ServiceNow Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Migrate Integration Artifacts Between Instances

Use this recipe to migrate integration artifacts (such as, integration flows, connections, lookups, libraries, agent groups, and so on) between two Oracle Integration instances.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe migrates integration artifacts (integration flows and associated resources) between two Oracle Integration instances. For example, you can use this recipe to migrate integration artifacts from the development environment to the testing environment.

The recipe performs the following tasks:

1. Checks the status of the source and target instances. If they are not reachable, the recipe terminates the migration process.
2. Executes a loop on the integration artifacts to migrate them if the instances are reachable.
3. Exports integration artifacts from the source instance to the target instance.
4. Sends the migration report to the configured email address.

System and Access Requirements

- Oracle Integration Source Instance
- Oracle Integration Target Instance

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Connection (Trigger)

1. In the Connections section, click the connection name.
2. In the Security section, select the Security Policy as either **OAuth2.0** or **Basic Authentication**.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Configure the Oracle REST Source Connection (Invoke)

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the Oracle Integration design time URL of the source instance. To obtain the design time URL, paste the service console URL in a new browser window. When the log in page of the Oracle Integration instance displays, note that the URL changes to a design time URL. All design time URLs start with the word design as shown in the following example: <code>https://design.integration.region.ocp.oraclecloud.com/?integrationInstance=NameOfServiceInstance</code> Enter only the following part of the URL: <code>https://design.integration.region.ocp.oraclecloud.com</code>

3. In the Security section, select the Security Policy as either **OAuth Client Credentials** or **OAuth Authorization Code Credentials**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle REST Target Connection (Invoke)

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.

Field	Information to Enter
Connection URL	Enter the Oracle Integration design time URL of the source instance. To obtain the design time URL, paste the service console URL in a new browser window. When the log in page of the Oracle Integration instance displays, note that the URL changes to a design time URL. All design time URLs start with the word design as shown in the following example: <code>https://design.integration.region.ocp.oraclecloud.com/?integrationInstance=NameOfServiceInstance</code> Enter only the following part of the URL: <code>https://design.integration.region.ocp.oraclecloud.com</code>

- In the Security section, select the Security Policy as either **OAuth Client Credentials** or **OAuth Authorization Code Credentials**.
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **Oracle_Integration_Artifacts_Migration_Utility** lookup table.

- In the Lookups section, click the lookup name.
- Enter the values for the following keys.

Key	Description
IntegrationStatusAPI	The API URL to check the status of the Oracle Integration instance.
EmailTo	The email address to which the email notifications are sent. Enter an email address of your choice.
EmailFrom	The email address from which the email notifications are sent. Enter an email address of your choice.
SourceIntegrationInstance	The source Oracle Integration service instance name.
TargetIntegrationInstance	The target Oracle Integration service instance name.

- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Test the recipe in Oracle Integration.

- a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
- b. On the Configure and run page, in the Request section, click **Body** and enter the payload with the following details:

Element	Value to Migrate Integrations
artifactType	Enter integrations.
code	Enter the integration identifier.
version	Enter the integration version. For example: 01.00.0000

For example:

```
{
  "artifact": [
    {
      "artifactType": "integrations",
      "code": "GETCURRENT",
      "version": "01.00.0000"
    }
  ]
}
```

- c. Click **Run**
The integration flow now automatically migrates the integration artifacts from the source instance to the target instance. The Activity Stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, under the **Body** tab, you'll find a HTTP status code as **202 accepted**.
3. Log in to your target instance and check for the migrated integration artifacts.

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)

Perform Arithmetic Operations

Use this recipe to perform basic arithmetic operations on the input data and handle faults encountered during the division operation.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

To use the recipe, you must install the recipe and configure the connection within the recipe. Subsequently, you can activate the integration flows of the recipe. Using the recipe, 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 recipe.
- Perform a division operation on the input data, and catch or handle faults encountered during the operation.

Integration	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: <pre><exam:AnyWithOption> <exam:option>mul</ exam:option> <exam:a>4</exam:a> <exam:b>12</exam:b> </exam:AnyWithOption></pre> Example Output Payload: <pre><env:Body> <AnyWithOptionResponse xmlns:wsa="http://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:msec="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></pre>

Integration	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: <pre><exam:Divide> <!--Optional:--> <exam:a>200</exam:a> <!--Optional:--> <exam:b>10</exam:b> </exam:Divide></pre> Example Output Payload: <pre><DivideResponse xmlns:wsa="http:// 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.IC SInstanceTrackingFunctions" 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_x path428770522" xmlns:orajs11="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path151171120" 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_x path1264540111" xmlns:orajs17="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path908143547" xmlns:orajs18="http://</pre>

Integration	Description
	<pre> www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path889320555" xmlns:soap12="http:// schemas.xmlsoap.org/wsdl/soap12/" xmlns:orajs19="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path226366833" xmlns:orajs12="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path754435699" xmlns:orajs9="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1372753807" xmlns:orajs13="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path248742258" xmlns:orajs7="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path591907397" xmlns:orajs14="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path2130408426" xmlns:orajs15="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1786525844" xmlns:orajs8="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path26387615" xmlns:orajs5="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1322872458" xmlns:orajs6="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path917958894" xmlns:orajs3="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path86288" xmlns:orajs4="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1996291323" xmlns:orajs1="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path729152639" xmlns:orajs2="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1611565945" xmlns:orajs0="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path503065307" xmlns:msc="http:// schemas.microsoft.com/ws/2005/12/wsdl/ </pre>

Integration	Description
	<pre>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"> <nstrgmpr:result>20.0</ nstrgmpr:result> </DivideResponse></pre>

Integration	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: <pre><exam:Divide> <!--Optional:--> <exam:a>60</exam:a> <!--Optional:--> <exam:b>0</exam:b> </exam:Divide></pre> Example Output Payload: <pre><DivideResponse xmlns:wsa="http:// 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.IC SInstanceTrackingFunctions" 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_x path908143547" xmlns:orajs31="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path889320555" xmlns:wsaw="http://www.w3.org/ 2006/05/addressing/wsdl" xmlns:wsa10="http:// www.w3.org/2005/08/addressing"</pre>

Integration	Description
	<pre> xmlns:orajs9="http://www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1197744749" xmlns:orajs7="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path605343633" xmlns:orajs8="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1851283090" xmlns:orajs5="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1886535281" xmlns:orajs6="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path640595824" xmlns:orajs3="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1073059868" xmlns:orajs4="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1024068758" xmlns:orajs1="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path729152639" xmlns:orajs2="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1496426531" xmlns:orajs0="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path503065307" 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_x path591907397" xmlns:orajs21="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path26387615" xmlns:orajs22="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1372753807" xmlns:orajs27="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path2130408426" xmlns:orajs28="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1786525844" xmlns:orajs29="http:// www.oracle.com/XSL/Transform/java/ </pre>

Integration	Description
	<pre> oracle.tip.pc.services.functions.JsExecutor_x path1264540111" xmlns:orajs23="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path428770522" xmlns:orajs24="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path151171120" xmlns:orajs25="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path754435699" xmlns:orajs26="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path248742258" xmlns:wsdl="http:// schemas.xmlsoap.org/wsdl/" xmlns:orajs10="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path48194708" xmlns:orajs11="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path2067245588" xmlns:orajs16="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1146925263" xmlns:orajs17="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1996291323" xmlns:orajs18="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1322872458" xmlns:soap12="http:// schemas.xmlsoap.org/wsdl/soap12/" xmlns:orajs19="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path917958894" xmlns:orajs12="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1611565945" xmlns:orajs13="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1822644184" xmlns:orajs14="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path133935528" xmlns:orajs15="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path86288" xmlns:nstrgdfl="http:// xmlns.oracle.com/procmon" xmlns:ns28="http:// xmlns.oracle.com/ics/tracking/ ics_tracking_context.xsd" xmlns:msc="http:// </pre>

Integration	Description
	<pre>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></pre>

Integration	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: <pre><exam:Divide> <!--Optional:--> <exam:a>10</exam:a> <!--Optional:--> <exam:b>0</exam:b> </exam:Divide></pre> Example Output Payload: <pre><DivideResponse xmlns:wsa="http:// 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.IC SInstanceTrackingFunctions" 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:nsmp0="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_x path428770522" xmlns:orajs11="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path151171120" xmlns:wsaw="http://www.w3.org/ 2006/05/addressing/wSDL" xmlns:wsa10="http://</pre>

Integration	Description
	<pre> www.w3.org/2005/08/addressing" xmlns:orajs16="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1264540111" xmlns:orajs17="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path908143547" xmlns:orajs18="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path889320555" xmlns:soap12="http:// schemas.xmlsoap.org/wsdl/soap12/" xmlns:orajs19="http://www.oracle.com/XSL/ Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path226366833" xmlns:orajs12="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path754435699" xmlns:orajs9="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1372753807" xmlns:orajs13="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path248742258" xmlns:orajs7="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path591907397" xmlns:orajs14="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path2130408426" xmlns:orajs15="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1786525844" xmlns:orajs8="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path26387615" xmlns:orajs5="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1322872458" xmlns:orajs6="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path917958894" xmlns:orajs3="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path86288" xmlns:orajs4="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1996291323" xmlns:orajs1="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x </pre>

Integration	Description
	<pre>path729152639" xmlns:orajs2="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path1611565945" xmlns:orajs0="http:// www.oracle.com/XSL/Transform/java/ oracle.tip.pc.services.functions.JsExecutor_x path503065307" 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"> <nstrgmpr:result>0</ nstrgmpr:result> </DivideResponse></pre>

System and Access Requirements

- Oracle Integration, Version 24.04 or higher

Before You Install the Recipe

To create a connection to the SOAP service, perform the configuration steps described in Create a SOAP Adapter Connection in *Using the SOAP Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle SOAP Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run details**.
 - b. In the **Run details** panel that opens, copy the **Metadata URL** value. This is the integration flow's endpoint URL.
 - c. From the external application such as SoapUI, load the WSDL hosted at the **Metadata URL** and send a request to the available operation along with the request message data.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify the results returned for the four integration flows of the recipe.
 - **SOAP Library Calculator Switch**: check if an arithmetic operation has been performed on the input data.
 - **SOAP Library Division Calculator**: check if a division operation has been performed on the input data.
 - **SOAP Library Global FaultHandler**: check if a division operation has been performed on the input data and a global fault handler is used to catch and handle errors.
 - **SOAP Library Div Inscope FaultHandler**: check if a division operation has been performed on the input data, and an in-scope fault handler is used to catch and handle errors.

Related Documentation

- [Using the SOAP Adapter with Oracle Integration 3](#)

Perform Multiple Operations on Resources

Use this recipe to perform multiple operations (HTTP actions) on different resource endpoints using a single integration flow.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how you can expose multiple entry points to a single, orchestrated integration flow that uses the REST Adapter as a trigger connection. You can configure each entry point with a different resource endpoint and operation (HTTP action), as necessary.

Using this recipe, you can send multiple, different REST requests from an external application to the same integration flow.

To use the recipe, you must install the recipe and configure the connection and other resources within it. Subsequently, you can activate the integration flow of the recipe and send multiple, different REST requests (with request parameters) to the integration flow's endpoint URL from an external application. When triggered, the integration flow performs the requested operations on the respective resources and returns the responses.

Note

As this is just a demonstration of how to use multiple HTTP verbs and resources in the REST Adapter trigger connection of an integration flow, the recipe contains the following customizations:

- Even though multiple resource endpoints are supported, this recipe uses a single, sample resource (*/emp*) and a sample sub-resource (*/emp/loc*).
- This recipe has the following sample operations configured:
 - modifyEmployee (PATCH)
 - updateEmployee (PUT)
 - getEmployee (GET)
 - getEmployeeLocation (GET)
 - deleteEmployee (DELETE)
 - addEmployee (POST)

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0, or higher

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).

2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send multiple REST requests to this endpoint URL along with the required request parameters.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page:
 - i. In the **Operation** field, select the required operation.
 - ii. In the Request section, enter the request parameters.
 - iii. Click **Run**.

The Activity stream pane appears displaying the status of the integration instance's execution.
 - c. In the Response section of the Configure and run page, you'll find a success response, 200 OK, and the response returned.
4. Check the responses returned to the source application.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

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.

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

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

Configure resources of the recipe. See [Configure Resources in a Package](#).

Related Documentation

- [Oracle Warehouse Management Cloud Integrating with Integration Cloud Services](#)
- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Post Slack Notifications for Completed SurveyMonkey Surveys

Use this recipe to post a notification message in Slack when a survey created with SurveyMonkey is completed by a respondent.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe posts notification messages to a specified Slack channel for completed SurveyMonkey surveys, according to a specified schedule in Oracle Integration. Using the recipe you can track survey responses by getting notification messages in Slack with details about completed survey responses.

The recipe uses the standard SurveyMonkey Adapter and the standard Slack Adapter. To use the recipe, you must install the recipe and configure connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered the integration queries for completed

surveys in SurveyMonkey, and posts notification messages for the completed surveys to the specified Slack channel.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39462) or higher
- SurveyMonkey
- An account in SurveyMonkey with the Administrator role
- Slack
- An account in Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SurveyMonkey instance and Slack instance in order to successfully connect to these external systems using Oracle Integration and post notification messages in Slack for completed survey responses.

Configure SurveyMonkey

You must perform the following prerequisite tasks in your SurveyMonkey instance.

1. Create and configure a SurveyMonkey app.
 - a. Log in to the SurveyMonkey Developer console at <https://developer.surveymonkey.com/>.
 - b. Click **My Apps** on the title bar of the SurveyMonkey Developer home page.
 - c. On the My Apps page, click **Add a New App**.
 - d. In the resulting APP CREATION dialog, enter the app's name in the **App Nickname** field, and select **Private App** as the **App Type** option.
 - e. Click **Create App**.

SurveyMonkey creates an app and displays the client ID and secret details on the resulting page.

- f. Under the Settings section, enter the following in the **OAuth Redirect URL** field.

`https://{OIC_HOST}:{OIC_SSL_PORT}/icsapis/agent/oauth/callback`

Replace OIC_HOST and OIC_SSL_PORT with specific values to your Oracle Integration instance.

- g. Get the SurveyMonkey API credentials.
 - i. Click **SETTINGS** to open the app's settings.
 - ii. Scroll down to the Credentials section, and note the values in the **Client ID** and **Secret** field.

Click the eye icon in the **Secret** field to view the value. You'll need the SurveyMonkey API credentials, that is the client ID and secret values while configuring the SurveyMonkey connection from Oracle Integration.

- h. Select the scopes.
 - i. In the Scopes section, select the following scopes:

- Create/Modify Surveys
 - Create/Modify Collectors
 - Create/Modify Responses
 - View Surveys
 - View Collectors
 - View Responses
 - View Response Details
- ii. Click **Update Scopes**.
You get a confirmation message that the scopes are successfully updated.
2. Create a survey in SurveyMonkey for which you want to receive responses.
- a. Log in to your SurveyMonkey account at <https://www.surveymonkey.com/user/sign-in/>.
 - b. On the SurveyMonkey Dashboard, click **CREATE SURVEY** on the title bar.
 - c. Create a survey using one of the following options in the Create a new survey pane on the left:
 - Start from scratch
 - Copy a past survey
 - Buy Targeted Response
 - Import Questions
 - Start from template
 - Build it for me
- Find information related to creating and managing surveys under the Design and Manage topic in the SurveyMonkey Help Center (<https://help.surveymonkey.com/>).
Note down the name of the survey that you have created.
- d. Click **Next** to preview your survey.
 - e. Click **Next** to view the COLLECT RESPONSES section.
 - f. Click **Send surveys your way**.
The COLLECT RESPONSES section displays the different ways in which you can collect the responses.
 - g. Click **Share a survey link**.
The COLLECT RESPONSES section displays the details of the web link collector.
Note down the name of the collector. For example, Web Link 1.

Configure Slack

Complete the following tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

1. Create a Slack app and configure it to receive notifications for successful Stripe charges.
 - a. Log in to the Slack app platform at <https://api.slack.com>.
 - b. From the workspace navigation menu, click your workspace.
 - c. Select **Administration** and then select **Manage apps**.
 - d. In the Slack app directory page, click **Build**.

- e. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
- f. In the Name app & choose workspace dialog, enter the following details.

Field	Information to Enter
App Name	Enter the name of your app. For example, Oracle Integration App.
Pick a workspace to develop your app in	Select your workspace from the drop-down list.

- g. Click **Create App**.
The app gets created and its Basic Information page appears.
- h. Get the Slack App API credentials. In the app's Basic Information page, scroll down to the App Credentials section. Note the client ID in the **Client ID** field, and the client secret in the **Client Secret** field.

These API credentials, that is the client ID and secret, were automatically generated when you created the app. You'll use these later while configuring connections to your Slack instance from Oracle Integration.
- i. Set permission and scope for your Slack App. In the left menu, under **Features**, select **OAuth & Permissions**.
- j. In the OAuth & Permissions page, go to the Redirect URLs section.
- k. Click **Add New Redirect URL** and enter the URL of your Oracle Integration instance. For example, `https://your_instance_URL:443/icsapis/agent/oauth/callback`, where `your_instance_URL` is the Host URL of your Oracle Integration instance.
- l. Click **Add**, and then click **Save URLs**.
- m. Scroll to the Scopes section, and under User Token Scopes click **Add an OAuth Scope**.
- n. Enter the following scopes:
 - channels:read
 - channels:write
 - chat:write
 - groups:read
 - groups:write
 - usergroups:write

Note the scopes. You'll use these later while configuring connections to your Slack instance from Oracle Integration.
- o. Press Enter.

2. You have to set up a workspace in Slack, create a channel in the workspace, and then add people who should receive the survey completed notification messages into the channel.

- a. Create a workspace in Slack.
 - i. Open the Slack get started page using the following URL:

`https://slack.com/get-started#/createnew`

- ii. Enter your email and click **Continue**.

- iii. Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.
 - b. Create a channel to post notifications.
 - i. On your Slack instance, select your workspace.
 - ii. Click the workspace name on the left navigation pane, and from the menu that displays, select **Create a channel**.
 - iii. In the Create a channel dialog, enter your channel's name in the **Name** field. For example, #survey-responses.
 - iv. Optionally, enter a suitable description in the **Description** field to let others know what the channel is about.
 - v. Click **Create**.
 - c. Add people to your channel.
 - i. On your Slack workspace, select the channel from the left navigation pane to open it.
 - ii. Click **Add people** on the top right of the channel.
 - iii. In the resulting Add people dialog, enter names, emails, or user groups of people you want to add in the field provided.
 - iv. Click **Add**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle SurveyMonkey Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID of the OAuth application created in SurveyMonkey. See Configure SurveyMonkey .
Client Secret	Enter the client secret of the OAuth application created in SurveyMonkey.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

You'll now be redirected to the SurveyMonkey login page.

6. Enter your SurveyMonkey account credentials and click **Log In**.

You are now displayed a page that prompts you to authorize the connection to the SurveyMonkey application created previously.

7. On the SurveyMonkey web page, click **Authorize**.
You're informed that access to SurveyMonkey is allowed. You can now switch back to the SurveyMonkey Connection window of Oracle Integration to test your connection.
8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
9. To return to the project workspace, click **Go back** .

Configure the Oracle Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Configure Slack .
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter the permission scopes you configured while creating and configuring the Slack app.

3. Click **Provide Consent**.
A new browser window opens to approve access to Slack.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration property values.
 - a. In the Integrations section of the project workspace, Hover over the **Oracle SurveyMonkey Slack Post Completed Responses** integration flow, then click **Actions**  and select **Update Property Values**.
 - b. In the Update Property Values dialog, note the properties that have been configured - **ChannelName**, **SurveyName**, and **NotificationMail**.
 - c. Click **ChannelName** and in the **New Value** field, enter the Slack channel name where you want the notification messages to be posted for completed survey responses.
Optionally, you can also enter the value of the Slack channel ID. To obtain the Slack channel ID, see [Configure Slack](#).

- d. Click **SurveyName** and in the **New Value** field, enter the name of the survey that you had created in Slack. See [Configure Slack](#).
 - e. Click **NotificationMail** and in the **New Value** field, enter the email ID where you want to notify in case a runtime error/exception occurs while running the integration.
 - f. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
3. Complete the survey.
 - a. Log in to your SurveyMonkey account.
 - b. Click **My Surveys** on the title bar.
 - c. Select the survey that you created.
 - d. Navigate to **COLLECT RESPONSES**, and use one of the collectors to participate in the survey.

For the purpose of testing, we'll choose the **Web link collector** option from the **ADD NEW COLLECTOR** drop-down list. Click the collector from the Survey Collector table, and in the resulting page that opens click **COPY** besides the web link URL. Copy and paste the link to a new browser window.
 - e. Access and complete the survey.
 4. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.
You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Check if a notification is posted in the Slack channel.
 - a. In Slack, navigate to the channel created for posting completed survey response notifications.
 - b. Check if the notification message for the survey that you completed in step 3 is posted in the channel.

Related Documentation

- [Using the SurveyMonkey Adapter with Oracle Integration 3](#)
- [Using the Slack Adapter with Oracle Integration 3](#)

Post Slack Notifications for New Marketo Leads

Use this recipe to post notification messages in Slack for new Marketo leads.

Topics:

- [About This Recipe](#)

- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate a Recipe or Accelerator](#)

About This Recipe

This recipe posts notification message to a specified Slack channel with details about new leads that are created in Marketo. It uses the standard Marketo Adapter and the Slack Adapter.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow manually or specify an execution schedule for it. When triggered, the integration flow queries for list of leads in Marketo. For the first time, it fetches all leads present in the Marketo instance, creates a message with the lead details (name and company), and posts the message to the specified Slack channel. For subsequent executions, it filters out the old leads and fetches only the new leads created since the last execution of the integration flow, and then posts notification messages to the specified Slack channel with details about the new leads.

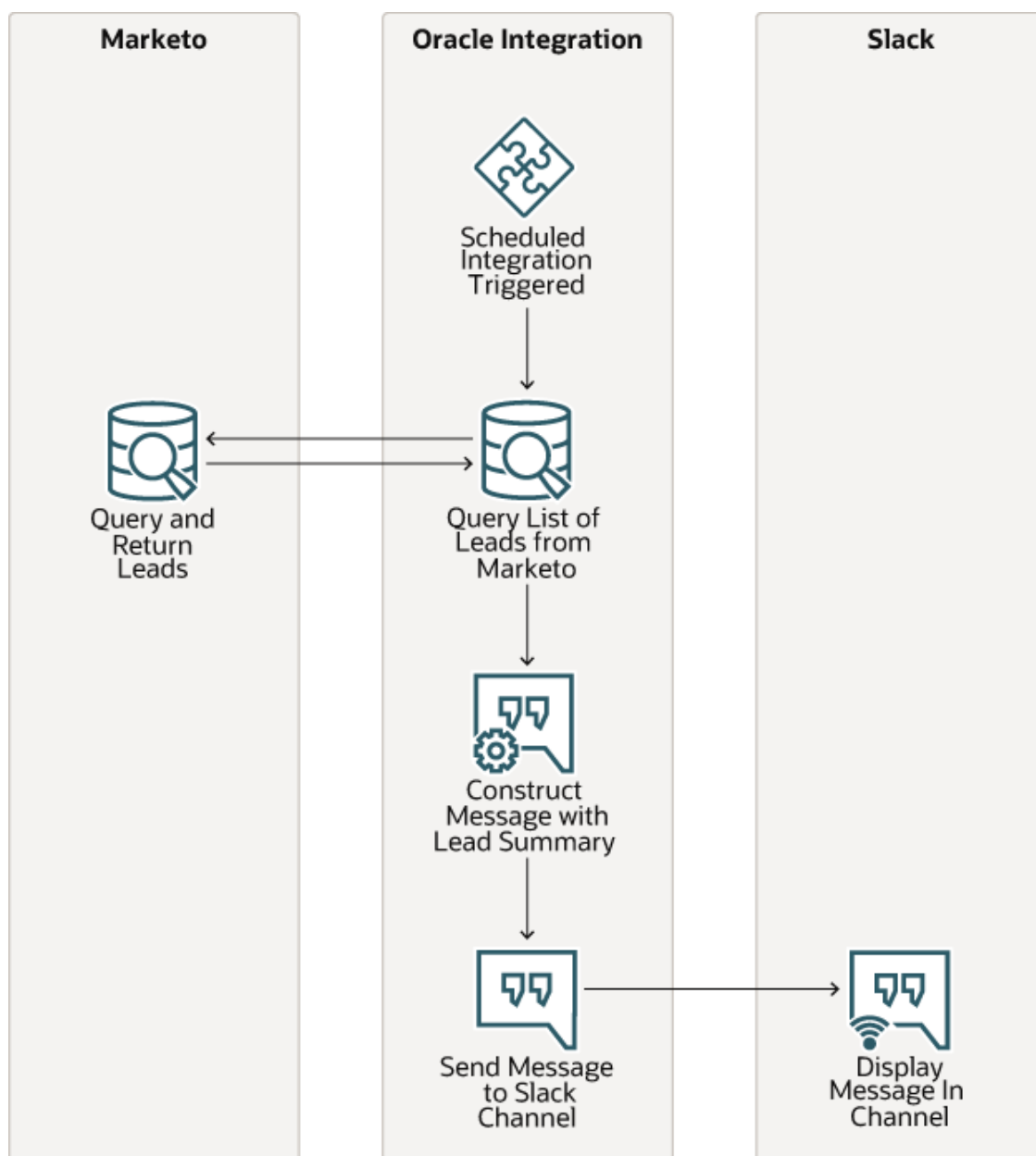
System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Marketo
- Slack
- An account in Marketo with the Administrator role
- An account in Slack with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.

When the integration flow of the recipe is triggered by an execution schedule or manual submission, it queries the Marketo instance for a list of leads. It filters out the old leads, that is, the leads that were already retrieved in the last integration flow; retrieves the new leads from the list, and then creates a message with details about the new leads. This message with details about the new leads is then sent to Slack and posted to the specified Slack channel.



Before You Install the Recipe

You must perform the following configuration tasks on your Marketo and Slack instances in order to connect to these external systems using Oracle Integration and successfully post notification messages in Slack for new Marketo leads.

Configure Marketo

Complete the following tasks in your Marketo instance to successfully connect to it from Oracle Integration.

1. Create an API only user role.
2. Create a customer service.
3. Obtain the Client ID and Client Secret.

4. Obtain the Munchkin ID.

For detail information on the above steps, see Prerequisites for Creating a Connection in *Using the Marketo Adapter with Oracle Integration 3*.

Configure Slack

Complete the following tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

Create and Configure a Slack App

1. Create a Slack App.
 - a. Log in to the Slack app platform at <https://api.slack.com>.
 - b. In the Slack API title bar, click **Your Apps**.
 - c. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
 - d. In the Name app & choose workspace dialog, enter the following details.

Field	Information to Enter
App Name	Enter the name of your app. For example, Oracle Integration App.
Pick a workspace to develop your app in	Select your workspace from the drop-down list.

- e. Click **Create App**.
The app gets created and its Basic Information page appears.
2. Get the Slack App API credentials.
 - a. In the app's Basic Information page, scroll to the App Credentials section.
 - b. Note the client ID in the **Client ID** field, and the client secret in the **Client Secret** field. These API credentials, that is the client ID and secret, were automatically generated when you created the app. You'll use these later while configuring connections to your Slack instance from Oracle Integration.
3. Set permission and scope for your Slack App.
 - a. In the left menu, under **Features**, select **OAuth & Permissions**.
 - b. In the OAuth & Permissions page, go to the Redirect URLs section.
 - c. Click **Add New Redirect URL** and enter the URL of your Oracle Integration instance. For example, https://your_instance_URL:443/icsapis/agent/oauth/callback, where your_instance_URL is the Host URL of your Oracle Integration instance.
 - d. Click **Add**, and then click **Save URLs**.
 - e. Scroll to the Scopes section, and under User Token Scopes click **Add an OAuth Scope**.
 - f. Enter chat:write.
 - g. Press Enter.

Set Up Your Slack WorkSpace and Channel

You have to set up a workspace in Slack, create a channel in the workspace, and then add people into the channel who should receive notification messages for events in the Amazon S3 bucket.

1. Create a workspace in Slack.
 - a. Open the Slack get started page using the following URL:

`https://slack.com/get-started#/createnew`
 - b. Enter your email and click **Continue**.
 - c. Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.
2. Create a channel to post notifications.
 - a. On your Slack instance, select your workspace.
 - b. Click the workspace name on the left navigation pane, and from the menu that displays, select **Create a channel**.
 - c. In the Create a channel dialog, enter your channel's name in the **Name** field. For example, #marketo-leads.
 - d. Optionally, enter a suitable description in the **Description** field to let others know what the channel is about.
 - e. Click **Create**.
3. Add people to your channel.
 - a. On your Slack workspace, select the channel from the left navigation pane to open it.
 - b. Click **Add people** on the top right of the channel.
 - c. In the resulting Add people dialog, enter names, emails, or user groups of people you want to add in the field provided.
 - d. Click **Add**.

Get the Slack Channel ID

You must get the channel ID of the channel that you created on your Slack workspace. You'll use the channel ID for configuring the integration properties in Oracle Integration, so that notifications are sent to the right Slack channel.

1. On your Slack workspace, navigate to the channel under **Channels** in the left navigation pane.
2. Right-click the channel name.
3. In the menu that displays, click **Copy link**.
4. Copy the link to a text editor, such as Notepad.
5. Note the nine characters after the last forward slash (/) in the link. This is the channel ID of the Slack channel. Note that the channel ID can be a combination of letters and numbers. For example, CB64YSB2D.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Marketo Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:
 - In the **Munchkin ID** field, enter the munchkin ID that you obtained earlier. See [Configure Marketo](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID obtained earlier. See Configure Marketo .
Client Secret	Enter the client secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Configure Slack .
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter chat:write:bot.

3. Click **Provide Consent**.

A new browser window opens to approve access to Slack.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions**  on the **Oracle Marketo Slack Schd New Lead Notification** integration flow, then select **Update property values**.

2. In the Update property values dialog, update the integration properties with appropriate values.

The recipe's integration flow contains the following properties. Click an integration property and update its value in the **New Value** field.

- a. **SlackChannelId**: This integration property holds the channel ID to which notifications are sent for new Marketo leads. Enter the channel ID of your Slack channel. See [Configure Slack](#).
 - b. **NumberOfLeadIds**: This integration property holds the total number of Marketo leads for which you want to send Slack notifications. Enter the number of lead information that you want to retrieve from Marketo. The default value is 10.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

Note

Before activating and running the recipe, ensure that there are new leads in your Marketo instance.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

- **LastExecutionDate**: This parameter stores the date of the most-recent successful run of the integration flow. For the initial run, it contains a default value. The parameter's value is automatically updated after each successful run, and only the Marketo leads created after the date stored as the parameter's **Current value** are processed by the integration in each run. If you want to change the date for a specific scenario, enter the date of your choice in the **New value** field in the format, yyyy-mm-dd.
- **NewestLeadId**: This parameter stores the Id of a Marketo lead. Only the Marketo leads created after the LeadId stored in this parameter are processed by the integration in each run. Enter a new LeadId of your choice from Marketo in the **New value** field.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if a notification message is posted in Slack.
 - a. In your Slack instance, navigate to the channel that you created for posting notifications for new Marketo leads.
 - b. Check if a notification message with the list of new leads from Marketo is posted to the channel.

Related Documentation

- *Using the Marketo Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

Post Slack Notifications for Oracle CPQ Quote Approvals

Use this recipe to post notifications in Slack for Oracle Configure, Price, and Quote Cloud (Oracle CPQ) quote approvals.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe posts a notification message to a specified Slack channel when an Oracle CPQ quote is sent for approval. The Slack notification message also contains a link to access the transaction approval page.

To use the recipe, you must install the recipe and configure connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe. When a quote is sent for approval in Oracle CPQ, the integration flow gets triggered, and it subsequently posts a notification message in the specified Slack channel with a link to the transaction approval page for the quote.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle CPQ
- Slack
- An account on Oracle CPQ with the Administrator role
- An account on Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle CPQ and Slack instances in order to successfully connect to these external systems using Oracle Integration and post Slack notifications for Oracle CPQ quote approvals.

Configure Oracle CPQ

To access Oracle CPQ using Oracle Integration and ensure that notifications are posted in Slack for quote approvals, you must perform certain configuration tasks in your Oracle CPQ instance.

Log in to your Oracle CPQ instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration with the **Web Services Only** permission.
 - a. On the Oracle CPQ home page, click **Admin Drawer** on the title bar.
 - b. In the Admin navigation pane, expand **Users** and click **Internal Users**.
The User Administration List page appears.
 - c. Add the user from the User Administration List page.
For details, see [Setting Up Users](#).
2. Obtain the REST Catalog URL. See Prerequisites for Creating a Connection.
3. Create an integration in Oracle CPQ to establish a connection from Oracle CPQ to Oracle Integration.
 - a. In the Admin navigation pane, expand **Integration Platform** and click **Integration Center**.
 - b. On the Integration Center page, click **Create Integration**.
 - c. From the **Type** drop-down list, select **Integration Cloud Service**.
 - d. In the resulting page enter the following details, and click Test to verify if your connection works.

Fields	Information to Enter
Name	Enter a suitable name.
Variable Name	Enter a variable name.
Discovery URL	Enter the Oracle Integration instance URL.
Username	Enter the username to access your Oracle Integration instance.
Password	Enter the password to access your Oracle Integration instance.

- e. Select **Enable Integration**, and then click **Save**.
4. Link the integration that you created to a process.
 - a. In the Admin navigation pane, expand **Commerce and Documents** and click **Process Definition**.
 - b. On the Processes page, select the process to which you want to link the integration. Select **Integrations** from the **Navigation** drop-down list, and click **List**.
 - c. On the resulting Integrations page, click **Add**.
 - d. In the Select Integration Types page, select **Integration Cloud Service** and click **Next**.
 - e. On the Edit Integration page, enter the following details and click **Add**.

Fields	Information to Enter
Name	Enter a suitable name.

Fields	Information to Enter
Variable Name	Enter a variable name.
Timeout	Enter the timeout in milliseconds.
Description	If required, enter a suitable description.
Action	Select the Export option.
Services	Select the Service.

5. Specify the action that triggers the integration.
 - a. On the Processes page, select the process linked to the integration. Select **Documents** from the **Navigation** drop-down list, and click **List**.
 - b. In the **Transaction** row of the Document List page, select **Actions** from the **Navigation** drop-down list, and click **List**.
 - c. On the Actions List page, select **Submit** and then **Request Approval**.
 - d. On the Admin Action page, click the **Integration** tab.
 - e. From the **Integration List** box, select the integration created in Step 3 and click the > arrow to add it to the **Selected Integration** box.
 - f. Click **Update**.
6. Specify the approval rule and the user that will approve the quote.
 - a. On the Processes page, select the process linked to the integration. Select **Integrations** from the **Navigation** drop-down list, and click **List**.
 - b. In the **Transaction** row of the Document List page, select **Actions** from the **Navigation** drop-down list, and click **List**.
 - c. On the Actions List page, select **Submit**.
 - d. Click the **General** tab, and navigate to **Approval Sequence** and **Edit User Approval**.
 - e. Click **+**, enter the **Reason Name** and **Variable Name**, and then click **Save**.
 - f. Select **Simple Condition**, specify the **Attribute Name**, **Operator**, and **Attribute Value**, and then click **Save**.
 - g. Click **Reason Flow**, click **Approvals**, and then select the approver who would approve the quote from the drop-down list.

Configure Slack

To access Slack using Oracle Integration, and ensure that Oracle CPQ quote approval notifications are posted in the specified Slack channel, you must perform certain configuration tasks in your Slack instance.

1. Create a Slack workspace.
 - a. Open the Slack get started page using the following URL:

`https://slack.com/get-started#/createnew`
 - b. Enter your email and click **Continue**.
 - c. Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.
2. Create a channel to post notifications.
 - a. On your Slack instance, select your workspace.

- b. Click the workspace name on the left navigation pane, and from the menu that displays, select **Create a channel**.
 - c. In the Create a channel dialog, enter your channel's name in the **Name** field. For example, #cpq-quote-approval.
 - d. Optionally, enter a suitable description for the channel in the **Description** field.
 - e. Click **Create**.
 3. Add people to your channel.
 - a. On your Slack workspace, select the channel from the left navigation pane to open it.
 - b. Click **Add people** on the top right of the channel.
 - c. In the resulting Add people dialog, enter names, emails, or user groups of people you want to add in the field provided.
 - d. Click **Add**.
 4. Create a Slack App.
 - a. Log in to the Slack app platform at <https://api.slack.com>.
 - b. In the Slack API title bar, click **Your Apps**.
 - c. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
 - d. In the Name app & choose workspace dialog, enter the following details:
 - i. In the **App Name** field, enter the name of your app.
For example, Oracle Integration App.
 - ii. In the **Pick a workspace to develop your app** in field, select your workspace from the drop-down list.
 - e. Click **Create App**.
The app gets created and its Basic Information page appears.
 5. Get the Slack App API credentials.
 - a. In the app's Basic Information page, scroll down to the App Credentials section.
 - b. Note the client ID in the **Client ID** field, and the client secret in the **Client Secret** field.
You'll use these later while configuring connections to your Slack instance from Oracle Integration.
 6. Set permissions for your Slack App.
 - a. In the left menu, under **Features**, select **OAuth & Permissions**.
 - b. In the OAuth & Permissions page, go to the Redirect URLs section.
 - c. Click **Add New Redirect URL** and enter the URL of your Oracle Integration instance.
For example, https://your_instance_URL:443/icsapis/agent/oauth/callback where *your_instance_URL* is the Host URL of your Oracle Integration instance.
 - d. Click **Add**, and then click **Save URLs**.
 - e. Scroll down to the Scopes section, and under User Token Scopes click **Add an OAuth Scope**.
 - f. Enter the following scopes:
 - channels:read
 - channels:write

- groups:read
- groups:write
- usergroups:write

Note the scopes. You'll use these later while configuring connections to your Slack instance from Oracle Integration.

- g. Press Enter.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle CPQ Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST Catalog URL . See Configure Oracle CPQ .
Connection URL	Enter the instance URL of Oracle CPQ.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration in Oracle CPQ. See Configure Oracle CPQ .
Password	Enter the password of the account created for Oracle Integration in Oracle CPQ.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID you obtained earlier while creating and configuring a Slack app. See Configure Slack .
Client Secret	Enter the client secret you obtained earlier while creating and configuring a Slack app.
Scope	Enter the permission scopes you configured while creating and configuring a Slack app.

3. Click **Provide Consent**.
A new browser window opens to approve access.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Update Integration Property

You have to update the integration property value so that Oracle CPQ quote approval notification messages are sent to the right Slack channel.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, click **SlackChannelId**.
3. In the **New Value** field, enter the name of the Slack channel where you want the notifications to be posted.
4. Click **Submit**.
You get a confirmation message that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle CPQ instance.
 - b. Create a transaction and specify the quote details.
Note that you can only submit a quote for approval, if it fulfills the specified approval rule.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if a notification message is posted in the Slack channel.
 - a. In your Slack instance, navigate to the channel you created for posting notifications for Oracle CPQ quote approvals.

- b. Check if a notification message with a link to the Oracle CPQ quote approval page is posted to the channel.
5. Check if you can access the Oracle CPQ quote approval page and approve the quote as the approver.
 - a. Click the link in the Slack notification message.
 - b. Log in to Oracle CPQ as the approver.
 - c. On the Approval Status page, enter Approved in the **My Approval** field. A green check mark indicates that the quote has been approved successfully.

The Approval History section displays details such as the user who approved the quote and the date on which the quote is approved.

Related Documentation

- *Using the Oracle CPQ Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

Post Slack Notifications for Oracle ERP Cloud Purchase Order Approvals

Use this recipe to post notification messages in Slack for Oracle ERP Cloud purchase order approvals.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe posts a notification message to a specified Slack channel when a purchase order is approved in Oracle ERP Cloud.

To use the recipe, you must install the recipe package and configure connections and other resources within the package. When a purchase order is approved in Oracle ERP Cloud, the recipe is triggered, and it posts a notification message about the purchase order approval to the specified Slack channel.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462), or higher
- Oracle ERP Cloud
- Slack
- An account on Oracle ERP Cloud with the Administrator role
- An account on Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance and Slack instance in order to successfully connect to these external systems using Oracle Integration and post notification messages in Slack for purchase order approvals in Oracle ERP Cloud.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information, see *Assign Required Roles to an Integration User* in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Slack

To access the Slack instance from Oracle Integration, and post notifications for approved Oracle ERP Cloud purchase orders, you must perform certain configuration tasks.

You have to set up your Slack instance by creating a workspace, creating a channel in the workspace to post notifications, and then adding people into the channel to whom you want to send the notification messages. For more information, refer to [Slack documentation](#).

In addition, you must create a Slack developer account, and then sign in to the Slack developer console at <https://api.slack.com> to perform the following tasks. For more information, see *Prerequisites for Creating a Connection* in *Using the Slack Adapter with Oracle Integration 3*.

1. Create an OAuth application.
2. Get the client ID and secret.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID you obtained. See Configure Slack .
Client Secret	Enter the client secret you obtained.
Scope	Enter the scope you specified. For example, <code>chat:write</code> .

3. Click **Provide Consent**.
A new browser window opens to approve access.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate value.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the following integration property with appropriate value.

- **slack_channel**: This integration property holds the name of the Slack channel to which purchase order approval notification messages are posted. Enter the name of the Slack channel.
3. Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle ERP Cloud instance with the Sales User role.
 - b. Click **Procurement** and then click **Purchase Requisition**.
 - c. Select any product and then add to cart.
 - d. Click **Submit**.
 - e. As the approver of the purchase order, approve the purchase order that was submitted.

You've successfully triggered the recipe. The recipe now posts a notification message in Slack about the purchase order approval.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In your Slack instance, check if a purchase order approval notification message is posted in the specified channel.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 flow 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 and configure the connections and other resources within the recipe. Then, you can activate and run some or all the integrations in the recipe, 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
- An account on ServiceNow with the Administrator role
- An account on Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Slack and ServiceNow instances in order to successfully connect to these external systems using Oracle Integration and post notification messages in 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

Complete the following tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

- [Set Up Your Slack Workspace and Channel](#)

- [Get the Slack Channel ID](#)
- [Create and Configure a Slack App](#)

Set Up Your Slack WorkSpace and 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, #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 notifications (#asset-creation).
 - A channel to post ServiceNow incident creation notifications (#incident-creation).
 - A channel to post ServiceNow request creation notifications (#request-creation).
 - A channel to post ServiceNow request closure notifications (#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** 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.
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.

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**.
 - c. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
 - d. In the Name app & choose workspace dialog, enter the following details.

Field	Information to Enter
App Name	Enter the name of your app. For example, Oracle Integration App.
Pick a workspace to develop your app in	Select your workspace from the drop-down list.

- e. Click **Create App**.
The app gets created and its Basic Information page appears.
2. Get the Slack App API credentials.
 - a. In the app's Basic Information page, scroll 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the ServiceNow Connection

1. In the Connections section, click the connection name.
2. In the Connection Properties section, enter the ServiceNow host name. For example: `https://instance_name.servicenow.com`. See [Configure ServiceNow](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Username	Enter your ServiceNow username. See Configure ServiceNow .
Password	Enter your ServiceNow password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Configure Slack .
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter the permission scopes you configured while creating and configuring a Slack app.

3. Click **Provide Consent**.

A new browser window opens to approve access to Slack.

4. Click **Allow**.

5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Update 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. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - 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. In the Update Property Values dialog, note the two properties that have been configured - **SlackChannelID** and **EmailNotification**.
3. 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 [Configure Slack](#).

4. 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.
5. Click **Submit**.

A message confirms that the properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Configure the basic authentication in ServiceNow for each integration flow of the recipe.
 - a. After activating the recipe, obtain the metadata URLs of all integration flows.
 - i. In the Integrations section of the project workspace, click **Actions**  on an integration flow, then select **Run details**.
 - ii. In the Run details panel, copy the **Metadata URL** value.

- b. Use each metadata URL to configure a separate basic authentication in ServiceNow. This is necessary to invoke the respective integration flow in Oracle Integration from ServiceNow.

See Configure Basic Authentication in *Using the ServiceNow Adapter with Oracle Integration*.

3. Run the recipe.

Log in to your ServiceNow | Developer instance as a system administrator and perform the following steps.

a. Create an asset in ServiceNow.

- i. On the left navigation pane, enter Asset in the Filter navigator field. The Self-Service menu displays all related modules.
- ii. In the Self-Service menu, under **Portfolios** click **All Assets**.
- iii. In the Assets window, click **New**.
- iv. In the resulting window, select the asset type.
- v. Fill the fields as required in the asset creation form.
- vi. Click **Submit**.

The new asset is created and displayed in the Assets window.

b. Create an incident in ServiceNow.

- i. In the navigation pane, enter Incident in the Filter navigator field. The Self-Service menu displays all related modules.
- ii. In the Self-Service menu, click **Incidents**.
- iii. In the Incidents window, click **New**.
- iv. Fill the fields as required in the incident creation form.
- v. Click **Submit**.

The new incident is created and displayed in the Incident window.

c. Create a request in ServiceNow.

- i. In the navigation pane, enter Request in the Filter navigator field. The Self-Service menu displays all related modules.
- ii. Scroll down the Self-Service menu, and under **Service Catalog** click **Requests**.
- iii. In the Service Catalog window, select a category. If required, select a sub-category.
- iv. Fill the fields as required in the request creation form.
- v. Click **Submit**.

d. Close a request in ServiceNow.

- i. In the navigation pane, enter Request in the Filter navigator field. The Self-Service menu displays all related modules.
- ii. Scroll down the Self-Service menu, and under **Service Catalog**, and then **Open Records**, click **Requests**. The Request window opens listing all open requests.
- iii. Select the request you want to close from the list.
- iv. In the **Approval** field, select either **Approved** or **Rejected**.
- v. In the **Request state** field, select an option from the drop-down list.

- vi. Click **Update**.
4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Check if the notification is posted in the following Slack channels.
 - a. Open the #asset-creation channel, and check if a notification message is posted for the asset you created in ServiceNow.
 - b. Open the #incident-creation channel, and check if a notification message is posted for the incident you created in ServiceNow.
 - c. Open the #request-creation channel, and check if a notification message is posted for the request you created in ServiceNow.
 - d. Open the #request-closure channel, and check if a notification message is posted for the request you closed in ServiceNow.

Related Documentation

- [Using the ServiceNow Adapter with Oracle Integration 3](#)
- [Using the Slack Adapter with Oracle Integration 3](#)

Post Slack Notifications for Stripe Charges

Use this recipe to post notification messages in Slack when a Stripe charge is successful.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe posts a notification message to a specified Slack channel when a Stripe charge is successful. The Slack notification message contains details of the payment such as amount, currency, customer name, product description, card type, and the last four digits of the card number. The notification message is in the following format: "A Payment of **amount currency** has been charged successfully for **customer name**'s order of item **product description** using a **card type** payment method ending with **card's last digit**."

To use the recipe, you must install the recipe and configure connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe. When triggered the integration retrieves the payment details from Stripe, and posts notification messages in the specified Slack channel.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- Stripe
- An account on Stripe with the Administrator role
- Slack
- An account on Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Stripe and Slack instances in order to successfully connect to these external systems using Oracle Integration and post notification messages in Slack for successful Stripe charges.

Configure Stripe

To access the Stripe instance from Oracle Integration, you must perform certain configurations on your Stripe instance.

1. Retrieve the API key associated with your Stripe account.
 - a. Log in to your Stripe instance as an administrator.
 - b. In the left navigation pane, click **Developers**, then **API keys**. The API keys page is displayed with all the keys associated with your account.
 - c. In the Standard keys section, click **Reveal test key** in the **Secret key** row.
 - d. Note the secret key.
2. Configure webhooks in your Stripe account. To register a webhook endpoint, see [Add a webhook endpoint](#).
 - In the **Endpoint URL** field, enter your Oracle Integration endpoint URL. For example, `https://<encodedUsername>:<encodedPassword>@<host>/ic/api/integration/v2/flows/rest/project/ ORCL-R-STRIPE_SLACK_CHARGE_NOTIF/ORCL-R-STRIPE_SLACK_CHARGE_NOTIF/1.0/`.
 - In the **Select events** field, select **charge.succeeded**.

Configure Slack

Complete the following tasks in your Slack instance to successfully connect to it using Oracle Integration and post notification messages.

1. Create a Slack app and configure it to receive notifications for successful Stripe charges.
 - a. Log in to the Slack app platform at <https://api.slack.com>.
 - b. From the workspace navigation menu, click your workspace.
 - c. Select **Administration** and then select **Manage apps**.
 - d. In the Slack app directory page, click **Build**.
 - e. Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
 - f. In the Name app & choose workspace dialog, enter the following details.

Field	Information to Enter
App Name	Enter the name of your app. For example, Oracle Integration App.
Pick a workspace to develop your app in	Select your workspace from the drop-down list.

- g. Click **Create App**.

The app gets created and its Basic Information page appears.

- [illegible]

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Stripe REST Trigger Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the following URL: https://api.stripe.com/ .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave API Key Based Authentication selected.
API Key	Enter the API key associated with your Stripe account. See Configure Stripe .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Configure Slack .
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter <code>chat:write:bot</code> .

3. Click **Provide Consent**.
A new browser window opens to approve access to Slack.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update integration property for the **Oracle Stripe Slack Charge Notification** integration flow. You can update the integration property only after activating this flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, specify the ID of the Slack channel in which the notifications must be posted. values. See [Configure Slack](#).
 - c. Click **Submit**.

A message confirms that the integration property has been updated successfully.

3. Make a payment through your Stripe account.
 - a. Log in to your Stripe account.
 - b. In the Stripe dashboard, navigate to the Payments tab.
 - c. Specify all the necessary details of the payment and click **Submit**.The recipe retrieves the payment details when the payment is successful.
4. Run the **Oracle Stripe Slack Charge Notification** integration flow.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Check if a notification is posted in the Slack channel.
 - a. In Slack, navigate to the channel configured for posting payment notifications.
 - b. Check if the notification message with the payment details is posted in the channel.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Slack Adapter with Oracle Integration 3*

Publish and Subscribe to Oracle Cloud Infrastructure Streaming Service

Use this recipe to publish and subscribe to the Oracle Cloud Infrastructure Streaming service.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe sends messages from Oracle Integration to Oracle Cloud Infrastructure Streaming, and receives messages from Oracle Cloud Infrastructure Streaming.

To use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe contains two integration flows - one to publish messages from

Oracle Integration to Oracle Cloud Infrastructure Streaming, and the other to subscribe to messages from Oracle Cloud Infrastructure Streaming.

- When the **Oracle Cloud Infrastructure Streaming - Publish** integration flow is triggered by a schedule, the recipe sends messages from Oracle Integration and publishes them to the specified stream pool in Oracle Cloud Infrastructure Streaming.
- When the **Oracle Cloud Infrastructure Streaming - Subscribe** is triggered by a schedule, the recipe consumes messages from the specified stream pool in Oracle Cloud Infrastructure Streaming, and publishes them to Oracle Integration.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Oracle Cloud Infrastructure Streaming

Before You Install the Recipe

You must perform certain configuration tasks on your Oracle Cloud Infrastructure Streaming service to successfully connect to it from Oracle Integration, and send and receive messages.

See Prerequisites for Creating a Connection in *Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Cloud Infrastructure Streaming Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the host and port to use to connect to Oracle Cloud Infrastructure Streaming in the **Bootstrap Servers** field. For example:
`streaming.region_location.oci.oraclecloud.com:port`.

Where the *region_location* is found in the URL when you log in to the Oracle Cloud Console and *port* is the port value (for example, 9092).

Note

Do not include `https://` in the **Bootstrap Servers** field.

3. In the Security section, enter the following details:

Field	Information to Enter
SASL Username	Enter the user name to access Oracle Cloud Infrastructure Streaming. This is a combination of the tenancy name, stream user name, and stream pool OCID separated by forward slashes (/). For example: xxxtenant/abcuser/yyystreampoolocid See Prerequisites for Creating a Connection in <i>Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3</i> .
SASL Password	Enter the password.
TrustStore (Optional)	Upload the trust store you generated. This field is optional.
TrustStore Password (Optional)	Enter the trust store password. This field is optional.

- Click **Save**. If prompted, click **Save** again.
- Configure an agent group for accessing the service hosted on your premises behind the fire wall. See *Configure an Agent Group in Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*.
- Test your connection to ensure that it is configured successfully. Click **Test** in the connection window. In the resulting window, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Integration Flow

To successfully publish messages to Oracle Cloud Infrastructure Streaming, you must edit the **PublishTo** endpoint in the **Oracle Cloud Infrastructure Streaming - Publish** integration flow.

- In the Integrations section, click the integration flow name.
- On the integration canvas, click the **Invoke** element, and then click **Actions**  and select **Edit** to edit the **PublishTo** 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 & Partition Page and Message Structure Page in Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*.

```
{
  "Id" : "123",    [ ns2:getFlowId () ]
  "Country": "US",
  "State": "MA" ,
  "County": "Worcester"
}
```

- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. View the Activity stream to check if messages were sent successfully to Oracle Cloud Infrastructure Streaming or if messages sent from Oracle Cloud Infrastructure Streaming were consumed successfully by Oracle Integration.

Related Documentation

- [Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3](#)
- [Oracle Cloud Infrastructure Streaming Service documentation](#)

Publish Shopify Orders to GCP Pub Sub

Use this recipe to publish Shopify orders to a topic in GCP Pub Sub.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe publishes Shopify orders to a topic in GCP Pub Sub, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads order records from the Shopify instance and, in turn, publishes them to a topic in GCP Pub Sub.

System and Access Requirements

- Oracle Integration, Version 28.2.0.0.0 or higher
- Shopify, Version 2020-01
- An account on Shopify with the Administrator role

- GCP Pub Sub
- An account on GCP Pub Sub with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Shopify and GCP Pub Sub instances in order to successfully connect to these external systems using Oracle Integration and import sales order details.

Configure Shopify

To configure Shopify, see Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Configure GCP Pub Sub

To successfully connect to GCP Pub Sub using Oracle Integration and publish orders, you must perform certain general configurations on your GCP Pub Sub instance. See Prerequisites for Creating a Connection in *Using the GCP Pub Sub Adapter with Oracle Integration 3*.

Also, you must create a topic to store the Shopify orders that are published. See [Create a Topic](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Shopify Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the Shopify host name. The Shopify host name appears in the Shopify home URL. For example: <code>https://hostname.myshopify.com</code> .
Shopify REST API Version	Enter 2020-01.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the API key obtained earlier. See Configure Shopify .
Password	Enter the password obtained earlier.

4. Click **Save**. If prompted, click **Save** again.

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the GCP Pub Sub Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the **project ID**. See [Configure GCP Pub Sub](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Google Client ID	Enter the client ID obtained earlier. See Configure GCP Pub Sub .
Google Client Secret	Enter the client secret obtained earlier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
 - a. In the resulting Sign in dialog, enter your Oracle Integration username and password, and click **Sign in**.
You'll now be redirected to the GCP Pub Sub login page.
 - b. Log in to your GCP Pub Sub account from the browser window that opens.
You are now displayed a page that prompts you to authorize the connection to the GCP Pub Sub application created previously.
 - c. On the GCP Pub Sub web page, click **ACCEPT**.
You're informed that access to GCP Pub Sub is allowed. You can now switch back to the GCP Pub Sub connection window of Oracle Integration to test your connection.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

- **orderCreatedAtorAfter**: Enter the date and time on or after which to retrieve orders. Use the following format: yyyy-MM-dd HH:mm.

- **pagelimit:** Enter the number of orders to retrieve per page. The maximum allowed value is 250.
 - b. On the Configure and run page, click **Run**.
- You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to GCP Pub Sub and perform the following steps.
 - a. Create a pull subscription. See [Create pull subscriptions](#).
 - b. Verify if you have received the subscription message for Shopify orders.

Related Documentation

- *Using the Shopify Adapter with Oracle Integration 3*
- *Using the GCP Pub Sub Adapter with Oracle Integration 3*

Read and Write Files on an FTP Server

This recipe copies a file from one directory to another on an FTP server.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe uses the FTP Adapter to read and write files to an FTP server. It enables you to copy a file from one directory to another on an FTP server.

With this recipe, you can also use the File Server, an embedded sFTP server within Oracle Integration. To use File Server, you must enable it for your Oracle Integration instance. See [Enable File Server in Using File Server in Oracle Integration 3](#).

System and Access Requirements

- Oracle Integration
- A secure FTP (sFTP) server or File Server
- An FTP client to access the external sFTP server

Before You Install the Recipe

Perform the following tasks on your sFTP server.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create an input directory (FTP Bulk Transfer) on the server from which to read files. The path of the directory should be: \FTP Bulk Transfer\.
3. Place a sample file (test.xml) in the input directory.
4. Create an output directory (FTP Write Folder) on the server to write files. The path of the directory should be: \FTP Write Folder\.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your FTP server and check if the sample file (test.xml) placed in the input directory has been copied to the output directory.

Related Documentation

- [Create an Integration to Import and Process Bulk Files](#)
- [Using File Server in Oracle Integration 3](#)

Resequence and Process Messages in a Specific Order

Use this recipe to resequence and process messages in a specific order.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe ensures that messages that enter a system are resequenced and processed in a specific order.

The Sequencing Problem

Often, you have a stream of requests that must be executed in order, for example, create account, update account address, and update account contacts. The later two activities can't occur until the first 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 recipe provides resequencing solution with the following:

- 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 out-of-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.
- Handles all errors, including system errors, network errors, and bad requests.

To use the recipe, you must install the recipe and configure the connections and other resources within it. The recipe 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 recipe to resequence different types of business integrations without modification to the integrations provided in the recipe.

- The resequencer **producer** integration is the source integration that receives the messages.
- The resequencer **producer** integration pushes the data to the database.
- The resequencer **consumer** integration pulls data based on scheduled frequency from the database.
- The resequencer **consumer** integration calls the **dispatcher** integration which then sends messages to the target system.
- The resequencer **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: <pre>\$ curl https:// my.integration.cloud/ic/api/ integration/v1/flows/rest/ RSQMANAGER/1.0/configs -v -u username:password</pre>

Operation	Path and Method	Description
Update config	Path: /configs/{type} Method: PUT	Update the config for the given type. Example of invocation: <pre>\$ 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</pre> config.json example: <pre>{ "maxConcurrent": 5, "timeWindow": 11 }</pre>
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: <pre>\$ 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"</pre>

System and Access Requirements

- Oracle Integration, 24.04 or higher
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Before You Install the Recipe

You have to complete certain configuration tasks on Oracle Autonomous Transaction Processing to successfully connect to it from Oracle Integration and resequence messages.

Complete the prerequisite tasks for creating a connection. See Prerequisites for Creating a Connection in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.

Additionally, as the recipe uses database tables to store the input messages, you must create the following database tables in Oracle Autonomous Transaction Processing.

1. Create table RSQ_CONFIG using the following command.

```
(
  TYPE          VARCHAR2(40 BYTE) NOT NULL,
  MAXCONCUR     NUMBER NOT NULL,
```

```

    TIMEWINDOW    NUMBER NOT NULL,
    CREATIONTIME  TIMESTAMP(3) NOT NULL,
    UPDATETIME    TIMESTAMP(3) NOT NULL,
    CONSTRAINT RSQ_CONFIG_PK PRIMARY KEY
    (
        TYPE
    )
);

```

2. Create table RSQ_GROUP using the following command.

```

(
    ID            VARCHAR2(40 BYTE) NOT NULL,
    TYPE          VARCHAR2(40 BYTE) NOT NULL,
    STATUS        VARCHAR2(10 BYTE) NOT NULL,
    CREATIONTIME  TIMESTAMP(3) NOT NULL,
    STATUSTIME    TIMESTAMP(3) NOT NULL,
    CONSTRAINT RSQ_GROUP_PK PRIMARY KEY
    (
        ID,
        TYPE
    )
);

```

3. Create table RSQ_MESSAGE using the following command.

```

(
    ID            VARCHAR2(40 BYTE) NOT NULL,
    GRP_ID        VARCHAR2(40 BYTE) NOT NULL,
    GRP_TYPE      VARCHAR2(40 BYTE) NOT NULL,
    SEQ_ID        NUMBER NOT NULL,
    STATUS        VARCHAR2(20 BYTE) NOT NULL,
    CREATIONTIME  TIMESTAMP(3) NOT NULL,
    STATUSTIME    TIMESTAMP(3) NOT NULL,
    PAYLOAD       CLOB NOT NULL,
    CONSTRAINT RSQ_MESSAGE_PK PRIMARY KEY
    (
        ID,
        GRP_ID,
        GRP_TYPE
    )
);

```

4. Create or replace procedure PROC_RSQ_CONFIG:

```

(
    GTYPE IN VARCHAR2,
    MAXCONCUR IN NUMBER,
    TIMEWINDOW IN NUMBER
)

```

as CC NUMBER

```

BEGIN
    SELECT COUNT(*) INTO CC FROM RSQ_CONFIG WHERE TYPE=GTYPE;

```

```

        IF (CC = 0) THEN
            INSERT INTO RSQ_CONFIG VALUES (GTYPE, MAXCONCUR, TIMEWINDOW,
            SYSDATE, SYSDATE);
        END IF;
    END PROC_RSQ_CONFIG;

```

5. Create or replace procedure PROC_RSQ_GROUP:

```

(
    GRP_ID IN VARCHAR2,
    GRP_TYPE IN VARCHAR2
)

as

GC NUMBER;
GS VARCHAR2(10)

BEGIN
    SELECT COUNT(*) INTO GC FROM RSQ_GROUP WHERE ID=GRP_ID AND
    TYPE=GRP_TYPE;
    IF (GC = 0) THEN
        INSERT INTO RSQ_GROUP VALUES (GRP_ID, GRP_TYPE, 'N', SYSDATE,
        SYSDATE);
    ELSE
        SELECT STATUS INTO GS FROM RSQ_GROUP WHERE ID=GRP_ID AND
        TYPE=GRP_TYPE;
        IF (GS = 'C') THEN
            UPDATE RSQ_GROUP SET STATUS='N' WHERE ID=GRP_ID AND
            TYPE=GRP_TYPE;
        END IF;
    END IF;
END PROC_RSQ_GROUP;

```

6. Create or replace procedure PROC_RSQ_GROUP_DONE:

```

(
    GID IN VARCHAR2,
    GTYPE IN VARCHAR2
)

as

MC NUMBER;
GS VARCHAR2(10);

BEGIN
    SELECT COUNT(*) INTO MC FROM RSQ_MESSAGE WHERE GRP_ID=GID AND
    GRP_TYPE=GTYPE;
    IF (MC = 0) THEN
        UPDATE RSQ_GROUP SET STATUS='C', STATUSTIME=SYSDATE WHERE ID=GID
        AND TYPE=GTYPE;
    ELSE
        UPDATE RSQ_GROUP SET STATUS='N', STATUSTIME=SYSDATE WHERE ID=GID

```

```
AND TYPE=GTYPE;  
    END IF;  
END PROC_RSQ_GROUP_DONE;
```

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the ATP DB RSQ Connection

1. In the Connections section, click the connection name.
2. In the Properties section, expand **Optional properties** and enter the service name in the **Service Name** field.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Upload the wallet zip file. See Before You Install the Recipe .
Wallet Password	Enter the password for the wallet.
Database Service Username	Enter the service user name of the database created for Oracle Integration on Oracle Autonomous Transaction Processing.
Database Service Password	Enter the password of the Database created for Oracle Integration on Oracle Autonomous Transaction Processing.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Trigger RSQManager Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the Security policy.
3. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Configure the REST Trigger EmployeeService Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the Security policy.
3. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Configure the REST RSQMessageConsumer Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the runtime URL of your Oracle Integration instance.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Provide the access token URI.
Client Id	Enter the client Id.
Client Secret	Enter the client secret.

4. Expand **Optional security**, and enter the following details.

Field	Information to Enter
Scope	Enter scope.
Client Authentication	Select Send client credentials as basic auth header .

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. Click **Save**. If prompted, click **Save** again.
7. To return to the project workspace, click **Go back** .

Configure the REST Employee Consumer Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the runtime URL of your Oracle Integration instance.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Provide the access token URI.
Client Id	Enter the client Id.
Client Secret	Enter the client secret.

4. Expand **Optional security**, and enter the following details.

Field	Information to Enter
Scope	Enter scope.
Client Authentication	Select Send client credentials as basic auth header .

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. Click **Save**. If prompted, click **Save** again.

7. To return to the project workspace, click **Go back** .

Configure the REST RSQProducer Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the runtime URL of your Oracle Integration instance.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Provide the access token URI.
Client Id	Enter the client Id.
Client Secret	Enter the client secret.

4. Expand **Optional security**, and enter the following details.

Field	Information to Enter
Scope	Enter scope.
Client Authentication	Select Send client credentials as basic auth header .

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. Click **Save**. If prompted, click **Save** again.
7. To return to the project workspace, click **Go back** .

Configure the Oracle REST OrderService Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the runtime URL of your Oracle Integration instance.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Provide the access token URI.
Client Id	Enter the client Id.
Client Secret	Enter the client secret.

4. Expand **Optional security**, and enter the following details.

Field	Information to Enter
Scope	Enter scope.
Client Authentication	Select Send client credentials as basic auth header .

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. Click **Save**. If prompted, click **Save** again.
7. To return to the project workspace, click **Go back** .

Configure the Oracle REST RSQDispatcher Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the runtime URL of your Oracle Integration instance.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Client Credentials .
Access Token URI	Provide the access token URI.
Client Id	Enter the client Id.
Client Secret	Enter the client secret.

4. Expand **Optional security**, and enter the following details.

Field	Information to Enter
Scope	Enter scope.
Client Authentication	Select Send client credentials as basic auth header .

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. Click **Save**. If prompted, click **Save** again.

7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Oracle REST REST EmployeeServiceFront** integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click **Body** and enter the following payload, then click **Run**.

```
{
  "Employee": {
    "hireDate": "1980-09-09",
    "name": "test5",
    "employeeId": "5",
    "department": "dept",
    "age": "31"
  }
}
```

- c. In the Integrations section of the project workspace, click **Actions**  on the **Oracle Scheduled RSQGroupConsumer** integration flow, then select **Run**.

- d. On the Configure and run page, click **Run**.

You've now successfully submitted the integrations for execution.

Note

You can also schedule these integrations to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. View the Activity stream to check if the input messages were resequenced and processed successfully.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Send a Hello Message

Use this recipe to send a message to an email address that is specified through a REST request from an external application.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how you can send a simple hello message through Oracle Integration to an email address received as request payload from an external application.

This is an application-driven recipe that sends a message when it receives a REST request with the name and email address of the user from an external application.

To use the recipe, you must install the recipe and configure the connections within the recipe. Subsequently, you can activate the integration flow of the recipe and send a REST request with the required payload (name and email ID) to the integration's endpoint URL from an external application.

When triggered, the recipe's integration flow sends a hello message to the email address specified in the request payload and returns a response to the external application.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0, or higher

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. From the external application, send a GET request to this endpoint URL along with the following request parameters:
 - **Name**: Specify the name of the user.
 - **Email**: Specify the email address to which the message must be sent.
3. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, under the **URI parameters** tab, enter the name of the user and specify the email address to which the message must be sent.
 - c. Click **Run**
The Activity stream pane appears displaying the status of the integration instance's execution.
 - d. In the Response section of the Configure and run page, you'll find a success status, 200 OK, and the response returned.
4. Check if a message is received on the email address specified in the request payload.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

Send Automatic Replies to Emails with a Specific Subject Using OCI AI Services

Use this recipe to send automatic replies to Microsoft Office 365 Outlook emails with a specific subject using the artificial intelligence capabilities of the following OCI AI services - Oracle Cloud Infrastructure Generative AI (OCI Generative AI) and Oracle Cloud Infrastructure Language (OCI Language).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe retrieves Microsoft Office 365 Outlook emails with specific subject and automatically sends responses to such emails by leveraging the artificial intelligence capabilities of OCI Language and OCI Generative AI services.

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe. The recipe contains three integration flows. The first integration flow **Oracle Get Email** runs at a defined schedule and fetches unread emails from the Outlook server to Oracle Integration. It filters emails with specific subjects from the retrieved emails and publishes them to the Publish event in JSON format. This in turn triggers the second integration flow **Oracle GenAI Quick ReplyToEmail** which subscribes to the Publish event. The second integration flow uses OCI Generative AI capabilities to generate a response to the email, which is then included in the body of the reply email. It also triggers the third integration flow **Oracle MaskingPIIfromEmail** that leverages OCI Language capabilities, specifically `BatchDetectLanguageEntities`, to identify key entities (for example: names, dates, locations, account numbers) in the email content, and masks sensitive data in the response. Subsequently, the AI generated response is sent to the email sender by the **Oracle GenAI Quick ReplyToEmail** integration flow.

System and Access Requirements

- Oracle Integration 3
- OCI Language
- OCI Generative AI

Before You Install the Recipe

You must perform certain configuration tasks on your Microsoft instance in order to successfully connect to it using Oracle Integration and send automatic responses to emails. Additionally, to use OCI Language and OCI Generative AI for sending automatic email responses with specific subject, you must complete certain prerequisites in Oracle Cloud Infrastructure Console.

1. Configure Microsoft Office 365 Outlook. See *Prerequisites for Creating a Connection in Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*.

2. Complete the prerequisites described in Prerequisites for OCI AI services in *Using Integrations in Oracle Integration 3*.

Want to learn more about OCI Language and OCI Generative AI? See the following topics in Oracle Cloud Infrastructure documentation:

- [Overview of OCI Language](#)
- [Batch Detect Language Entities API](#)
- [Personal Identifiable Information](#)
- [Overview of OCI Generative AI](#)

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Microsoft Outlook Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client Id.
Client Secret	Enter the client secret.
Scope	Enter the scope URL. For example <code>https://graph.microsoft.com/Mail.ReadWrite</code> <code>https://graph.microsoft.com/Mail.Send offline_access</code>

3. Click **Provide Consent**. This enables Oracle Integration to interact with the Microsoft Office 365 Outlook account used to create the application at `https://portal.azure.com/`. If everything is correct, you are prompted for the Oracle Integration credentials.
 - a. Enter the credentials and click **OK**. These are the same credentials you use to log in to Oracle Integration.
 - b. Enter the Microsoft Office 365 Outlook account credentials.
A page is displayed asking for permission to interact with the account.
 - c. Click **Yes**.
The Access Allowed! page is displayed.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Integration Flows

Edit the integration flows to configure actions for the following OCI AI services.

- [OCI Language](#)
- [OCI Generative AI](#)

OCI Language

1. In the Integrations section, click the **Oracle MaskingPIIfromEmail** integration flow.
2. On the integration canvas, click the **OCI Language** element, and then click **Actions . . .** and select **Edit**.
3. Click **Action** and select **Personal identifiable information (PII)**, and then click **Continue**.
4. From the **Compartment** drop-down list, select the OCI compartment.
5. Click **Next** and then **Done**.
6. Click **Save** to save the integration.
7. To return to the project workspace, click **Go back** .

To know more about the language action, see Perform Text Analysis and Translation with a Language Action in *Using Integrations in Oracle Integration 3*.

OCI Generative AI

1. In the Integrations section, click the **Oracle GenAI Quick ReplyToEmail** integration flow.
2. On the integration canvas, click the **OCI Generative AI** element, and then click **Actions . . .** and select **Edit**.
3. From the **Select Categories** drop-down list, select **On Demand**.
4. Click **Action** and select **Chat**, and then click **Continue**.
5. Select the OCI region, OCI compartment, model, and model ID.
6. Click **Next** and then **Done**.
7. Click **Save** to save the integration.
8. To return to the project workspace, click **Go back** .

To know more about the generative AI action, see Generate Text with a Generative AI Action in *Using Integrations in Oracle Integration 3*.

Configure the Lookup Tables

Edit the values of the lookup keys as required in the recipe's lookup tables.

The recipe contains the following lookup tables:

- **AILanguageParameters**: This lookup table holds input parameters for OCI Language.
 - **GenerativeAIParameters**: This lookup table holds input parameters for OCI Generative AI.
1. In the Lookups section, click the lookup name.
 2. Edit the lookup table.
 - In the **AILanguageParameters** lookup table, enter required values to prepare request for BatchDetectLanguageEntities API in the **Oracle MaskingPIIfromEmail** integration flow.
 - In the **GenerativeAIParameters** lookup table, enter required values to prepare request for Generative AI API in the **Oracle GenAI Quick ReplyToEmail** integration flow.
 3. Click **Save**. If prompted, click **Save** again.
 4. To return to the project workspace, click **Go back** .

Update Integration Properties

Update integration property for the **Oracle Get Emails** integration flow.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, enter the value for the **MailSubjectToRead** integration property. For example: Account Payment Dispute.

This integration property holds the value for the email subject on the basis of which emails retrieved from the Outlook server are filtered.

3. Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle Get Email** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The integration now retrieves unread emails from the Outlook server and filters them based on a specific subject (for example: *Account Payment Dispute*) for further processing. Subsequently the second and third integration flows get triggered and automatic responses (without any manual interventions) are sent to emails with the specific subject.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check your Microsoft Office 365 Outlook email inbox if you got automatic replies to emails that you had sent with the specific subject (*Account Payment Dispute*).

Open the response and verify if sensitive information, such as bank account number, have been masked in the email.

Related Documentation

- *Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*
- [OCI Language](#)
- [OCI Generative AI](#)

Send Compliance Documents from DocuSign to New ServiceNow Users

Use this recipe to send an IT compliance document from DocuSign to a new ServiceNow user.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe sends an IT compliance document from DocuSign to the email address of a new IT service management user, when the user is created in ServiceNow. The integration uses the standard ServiceNow Adapter and the standard DocuSign Adapter.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Then you can activate and run the recipe manually. When a new user is created in ServiceNow, the integration flow gets triggered, and it sends the IT compliance document from DocuSign to the user's email address specified in the new user record form in ServiceNow. The user electronically signs the compliance document received in email and completes the process.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- ServiceNow
- DocuSign
- An account in ServiceNow with the Administrator role

- An account in DocuSign with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your ServiceNow and DocuSign instances in order to connect to these external systems using Oracle Integration and successfully send compliance documents.

Configure ServiceNow

You can use admin account credentials or custom account credentials while configuring the ServiceNow connection in Oracle Integration.

If you do not want to use an admin account, create a custom integration user and assign the required role and permissions to the user. For information about creating a custom user, see *Prerequisites for Creating a Connection in Using the ServiceNow Adapter with Oracle Integration 3*.

Configure DocuSign

You must complete the following prerequisite tasks to connect to DocuSign from Oracle Integration, and send compliance documents to new ServiceNow users.

1. [Create an App and Integration Key](#)
2. [Create a Template](#)
3. [Get the Template ID](#)

Create an App and Integration Key

You must create an app and get information such as the integration key, that is the client ID, and the client secret to successfully connect to DocuSign from Oracle Integration.

1. Log in to your DocuSign Developer account at <https://appdemo.docusign.com/home>.
2. On the DocuSign home page, click **Settings** on the title bar.
3. In the left navigation menu, scroll down to **INTEGRATIONS**, and click **Apps and Keys**.
4. On the Apps and Keys page, click **ADD APP AND INTEGRATION KEY**.
5. In the Add Integration Key dialog, enter a name for the app in the **App Name** field, and click **CREATE APP**.
6. On your app's page, under General Info, note the **Integration Key**. Click the **Copy to clipboard** icon to copy the integration key value.

You'll need the integration key (client ID) value while configuring the DocuSign connection from Oracle Integration.

7. In the Authentication section:
 - a. Under **User Application**, select **Authorization Code Grant**.
 - b. Under **Secret Keys**, click **ADD SECRET KEY**
 - c. Click the **Copy to clipboard** icon to copy the secret key value.
 - d. Note the secret key value. You'll need it later while configuring the DocuSign connection from Oracle Integration.

8. In the Additional settings section, under **Redirect URIs**, click **ADD URI** and enter your redirect URI.

`https://{OIC_HOST}:{OIC_SSL_PORT}/icsapis/agent/oauth/callback`

Replace OIC_HOST and OIC_SSL_PORT with specific values to your Oracle Integration instance.

9. Click **Apps and Keys** to go back to the Apps and Keys page. The newly created app is displayed under **Apps and Integration Keys**.
10. Get the **API Account ID**.
 - a. Under My Account Information of the Apps and Keys page, copy the **API Account ID**. Click the **Copy to clipboard** icon.
 - b. Note the value of the **API Account ID**. You'll need it later while configuring the DocuSign connection from Oracle Integration.
11. Get the DocuSign **Account ID**.
 - a. Click the profile image on the upper-right side of the page.
 - b. On the menu that displays, note the Account ID that is under name or organization name. You'll need it later while configuring the DocuSign connection and also for updating the integration properties in Oracle Integration.

Create a Template

The compliance document that is sent to new ServiceNow users from DocuSign must be based on a pre-defined template. You have to create and set up this template in DocuSign.

1. Log in to your DocuSign account at <https://account.docusign.com>.
2. On the DocuSign home page, click **Templates** on the title bar.
3. In the left navigation pane, click **New**, and from the drop-down menu click **Create Template**.
4. In the resulting window, enter a name and description for the template.
5. In the **Add Documents** section:
 - a. To add a file from your local storage, click **UPLOAD**.
 - b. To add a file from one of the authorized cloud storage providers click **GET FROM CLOUD**, and choose an option from the drop-down menu. The available options are Box, DropBox, GoogleDrive, and OneDrive.

See the topic Supported File Formats in the *DocuSign eSignature User Guide*.

6. In the **Add Recipients** section:
 - a. Enter the role of the recipient in the **Role** field.
 - b. Enter the name of the recipient in the **Name** field.
 - c. Enter the email of the recipient in the **Email** field.
 - d. To add more recipients, click **ADD RECIPIENT**, and repeat the steps above.
 - e. You can also enable bulk send for multiple recipients.

See the topic Bulk Send for Multiple Recipients in the *DocuSign eSignature User Guide*.

- f. Optionally, you can set a routing method or add an authentication method.

To learn more, see *Add Recipients* in the *DocuSign eSignature User Guide*.

7. Scroll down to the **Message to All Recipients** section, and enter a standard email subject and message that you want to send to all the recipients of your template.

See the topic *Add Messages* in the *DocuSign eSignature User Guide*.

8. If you don't want to add any more fields to the template, click **SAVE AND CLOSE**.
9. If you want to add fields, click **NEXT** and follow the steps described in *Add Fields to Documents* in the *DocuSign eSignature User Guide*.
10. Click **SAVE AND CLOSE**.

Tip: To quickly access a topic in the *DocuSign eSignature User Guide*, you can search for the topic in the DocuSign | Support page (<https://support.docusign.com/en/home>).

Get the Template ID

You have to get the ID of the predefined template that you created in DocuSign and pass the template ID value to the integration properties so that the right compliance document is sent to new ServiceNow users.

1. Log in to your DocuSign account at <https://account.docusign.com>.
2. Click **Templates** on the title bar of the home page.
3. Search for your template by entering your template's name in the Search field.
4. Click the template to open it in the detail view.
5. In the template's detail view, click the **Template ID** link.

6. In the resulting Template ID dialog, click **COPY** to copy the template ID.

You get a message that the template ID has been copied to the clipboard.

7. Note the template ID. You'll need it later while updating the integration properties from Oracle Integration.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle DocuSign Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID (integration key) that you obtained earlier while creating an app and integration key in DocuSign. See Create an App and Integration Key .
Client Secret	Enter the client secret that you obtained earlier while creating an app and integration key in DocuSign.
Scope	Enter <code>signature extended</code> .

Field	Information to Enter
Instance Type	Enter sandbox.
Account ID	Enter the DocuSign Account ID followed by the API Account ID you obtained while creating an app and integration key in DocuSign.

- 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.
- Enter your DocuSign account credentials, then click **Authorize** and grant access.
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle ServiceNow Connection

- In the Connections section, click the connection name.
- In the Connection Properties section, enter the ServiceNow host name. For example:
`https://instance_name.servicenow.com`.
- In the Security section, enter the following details:

Field	Information to Enter
Username	Enter your ServiceNow username.
Password	Enter your ServiceNow password.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Update the Integration Property

You must update the integration property values so that the right compliance document is sent to a new IT user in ServiceNow, and run time exception emails are sent to the right email ID.

- In the Integrations section, click **Actions ...** on the **Oracle ServiceNow DocuSign User Doc SignIn** integration flow, then select **Update property values**.
- In the Update Property Values dialog, note the three properties that have been configured - DocuSignAccountId, DocuSignTemplateId, and NotificationMail.
- Click **DocuSignAccountId**, and in the **New Value** field, enter your DocuSign Account ID. You obtained the DocuSign Account ID while creating an app and integration key. See step 11 in [Create an App and Integration Key](#).

4. Click **DocuSignTemplateId**, and in the **New Value** field, enter the Template ID that you obtained earlier. See [Get the Template ID](#).
5. Click **NotificationMail**, and in the **New Value** field, enter the email ID where you want to notify in case a run time exception/error occurs while running the integration.
6. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Configure the basic authentication in ServiceNow for the integration flow of the recipe.
 - a. After activating the recipe, obtain the metadata URL of the integration flow.
 - i. In the Integrations section of the project workspace, click **Actions** . . . on the integration flow, then select **Run details**.
 - ii. In the Run details panel, copy the **Metadata URL** value.
 - b. Use the metadata URL to configure a basic authentication in ServiceNow. This is necessary to invoke the integration flow in Oracle Integration from ServiceNow.
See *Configure Basic Authentication in Using the ServiceNow Adapter with Oracle Integration*.

3. Run the recipe.

Create a new user in ServiceNow.

- a. Log in to your ServiceNow | Developer instance as a system administrator.
- b. In the navigation pane, enter **User Administration** in the Filter navigator field.
The Self-Service menu displays all related modules.
- c. In the Self-Service menu, under **User Administration** click **Users**.
- d. On the Users page, click **New**.
- e. Fill in the required information in the new user record form.

Field	Information to Enter
User ID	Enter a unique identifier for the new user's ServiceNow log in user name.
First Name	Enter the user's first name.
Last Name	Enter the user's last name.
Department	Select the user's department from the look up list.
Title	Select the user's job title. Click the Suggestion icon to select from a list of suggested titles.
Email	Enter the user's email address. Note that this is the email address to which the compliance document will be sent to the user.

- f. Click **Submit**.

The new user record appears at the top of the list in the Users page.

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Check if you received the compliance document from DocuSign in the specified email address, and you're able to access and sign it.
 - a. Sign in to the email account that you specified in the ServiceNow new user form in step 3e.
 - b. Check if you received an email to review and sign the compliance document.
 - c. Open the email, and click **REVIEW DOCUMENT**.
 - d. Select the **I agree to use Electronic Records and Signatures** check box.
 - e. Select **Continue** to access the compliance document.
 - f. On the compliance document, click the **START** tag on the left. You're taken to the Signature field of the document. Optionally, drag the **START** tag to the part of the document where you want to sign.
 - g. Click the **SIGN** tag.
 - h. In the resulting Adopt Your Signature dialog, verify if your name and signature is correct, and then click **ADOPT AND SIGN**.

If needed, you can change the information.

 - Enter your name in the **Full Name** field.
 - Enter your initials in the **Initials** field.
 - i. On the compliance document, click **FINISH**.
 - j. In the dialog that lets you know that a copy of the signed document will be sent to your email address, click **Continue**.

You get an email with the signed compliance document.

Related Documentation

- *Using the ServiceNow Adapter with Oracle Integration 3*
- *Using the DocuSign Adapter with Oracle Integration 3*

Send Files from OCI Object Storage to Oracle Autonomous Transaction Processing

Use this recipe to send files from OCI Object Storage to Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe transfers files from OCI Object Storage to Oracle Autonomous Transaction Processing (ATP), whenever a file is uploaded in OCI Object Storage.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a new object (file) is uploaded to a bucket in OCI Object Storage, the integration flow is triggered, and it downloads the file from OCI Object Storage. Further, the recipe sends the downloaded file to ATP as a Binary Large Object (BLOB) attachment.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 (210129.2200.39462), or higher
- OCI Object Storage
- An account on OCI Object Storage with the Administrator role
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your OCI Object Storage and Oracle Autonomous Transaction Processing instances in order to successfully connect to these external systems using Oracle Integration and transfer files.

Configure Oracle Database Cloud Service

To access the Oracle Database Cloud Service using Oracle Integration, you must perform certain general configurations on your Oracle Database Cloud Service instance.

See Prerequisites for Creating a Connection in *Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*.

Also, perform the following tasks to configure Oracle Database Cloud Service for this recipe.

1. Create a schema named OBJECTSTORE_ATP. See [Creating Schemas](#).
2. Run the following script to create a table named FILE_TABLE under the schema.

```
CREATE TABLE OBJECTSTORE_ATP.FILE_TABLE
(
  FILE_NAME      VARCHAR2 (128),
  CREATION_DATE  DATE,
  FILE_BLOB      BLOB
)
```

Configure OCI Object Storage

To access OCI Object Storage instance from Oracle Integration, you must create a bucket and create a set of API signing keys in your OCI Object Storage instance. See [How to use the OCI Object Storage from the Oracle Integration Cloud](#).

Also, ensure that the **Emit Object Events** is enabled for the bucket created.

Configure OCI Notification Service

To configure OCI events and notification services, see [Configuring OCI Events and Notification Services for OCI Data Integration \(OCI-DI\)](#).

Note the following while configuring notification services:

- While creating the subscription, create a HTTPS subscription. In the Create Subscription page, select HTTPS as protocol.
- In the URL field, specify your endpoint URL in the following format: `https://username:password@<host> /ic/api/integration/v2/flows/rest/project/ORCL-R-STORE_ATP_FILETRANSFER/ORACLE_OBJECTST_ATP_FILETRAN/1.0/sendcreateobjectevent`

Specify the following values while configuring events.

- In the rule conditions section, specify the following values.

Field	Information to Enter
Condition	Select Event Type .
Service Name	Select Object Storage .
Event Type	Select Object – Create .

- In the Attribute(filter) section, specify the following values.

Field	Information to Enter
compartmentName	Enter a desired compartment. This filters events from a specific compartment.
bucketName	Enter a desired Object Storage Bucket. This filters events only in the Bucket from where you want to source files for your OIC flow.
Event Type	Select Object – Create .

- In the Actions section, specify the following values.

Field	Information to Enter
Action Type	Select Notifications .
Notifications Compartment	Enter a desired compartment. This filters notifications from a specific compartment.
Topic	Select the topic created previously.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.

Field	Information to Enter
Connection URL	Enter the object storage endpoint that matches your location from the <i>API Endpoints</i> list. See Object Storage Service API .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave OCI Signature Version 1 selected.
Tenancy OCID	Enter your tenancy OCID. See Finding Your Tenancy OCID (Oracle Cloud Identifier) .
User OCID	Enter your user OCID. See Resource Identifiers .
Private Key	Enter the private key. See <i>How to Generate API Signing Key</i> in Required Keys and OCIDs . Ensure that the key pair is in PEM format. Also, upload public key to the Oracle Cloud Console. See <i>To upload or paste API key</i> in Required Keys and OCIDs .
Finger Print	Enter your finger print. See <i>How to get the Key's Fingerprint</i> in Required Keys and OCIDs .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle DBaaS Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Oracle Wallet .
Wallet	Enter the username of your Oracle Wallet. See Configure Oracle Database Cloud Service .
Wallet Password	Enter the password of your Oracle Wallet.
Database Service Username	Enter the username of your database service.
Database Service Password	Enter the password of your database service.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Update the Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, update the following properties:
 - **FromAddress**: This integration property holds the email address from which error notification emails are sent. Enter an email address of your choice.
 - **ToAddress**: This integration property holds the email address to which error notification emails are sent. Enter an email address of your choice.
3. Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe. To do so, upload an object (file) to a bucket in OCI Object Storage. See [Uploading Objects to a Bucket or Folder](#).

The Activity Stream pane appears displaying the status of the integration instance's execution.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if the file has been transferred to ATP. See [List Contents of Directory in Autonomous Database](#).

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Send Files from Your Application by Email

Use this recipe to send files from an external application to a specified email address.

Topics:

- [About This Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

When this recipe receives a REST request containing a file from an external application or from another integration flow in Oracle Integration, it 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 and configure the connection and other resources within the recipe. Subsequently, you can activate the integration flow of the recipe 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.

Note

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

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and enter the recipient email address to which you want the files sent.

1. In the Lookups section, click the lookup name.
2. Against the **SenderEmail** and **RecipientEmail** fields, enter the corresponding email addresses.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.

- c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
- d. From the external application, send a POST request to this endpoint URL along with the 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. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click **Body** and upload a file from your computer.
 - c. Click **Run**

The integration flow 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.
 - d. In the Response section of the test page, you should see the status as **202 Accepted**.
4. 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 3](#)

Send New Appointment Details from an EMR Application to a Healthcare Provider

Use this recipe to send patient details via an email to a healthcare provider when a new appointment is made in an Electronic Medical Records (EMR) application.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

When a new appointment is made for a patient in an EMR application, the recipe receives an event, and subsequently sends an email notification to the healthcare provider with the patient details.

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe. The recipe captures the event that occurs in the EMR application using the MLLP adapter, translates the incoming message into Oracle Integration readable format using the **Healthcare** action, extracts the required data from the HL7 message, and finally sends the data over email to the healthcare provider.

The recipe contains the **New Appointment Notification to Care Provider** integration flow that receives the inbound HL7 messages from the EMR application when a new appointment is made. In this example, the EMR application sends the **SIU_S12 (Notification of new appointment booking)** message.

System and Access Requirements

- Oracle Integration 3
- Oracle Integration for Healthcare Support for HL7

Before You Install the Recipe

To successfully send email notifications with patient details to a healthcare provider when a new appointment is made in an EMR application, you have to complete the following prerequisite tasks.

1. Create an agent group

You must create an agent group in Oracle Integration before you can run the connectivity agent installer. Creating the agent group automatically creates an OAuth client application in Oracle Identity Cloud Service. This enables the connectivity agent to use OAuth 2.0 token-based authentication when invoking Oracle Integration endpoints.

To create an agent group, see *Create an Agent Group in Using Integrations in Oracle Integration 3*.

2. Download and install the connectivity agent.

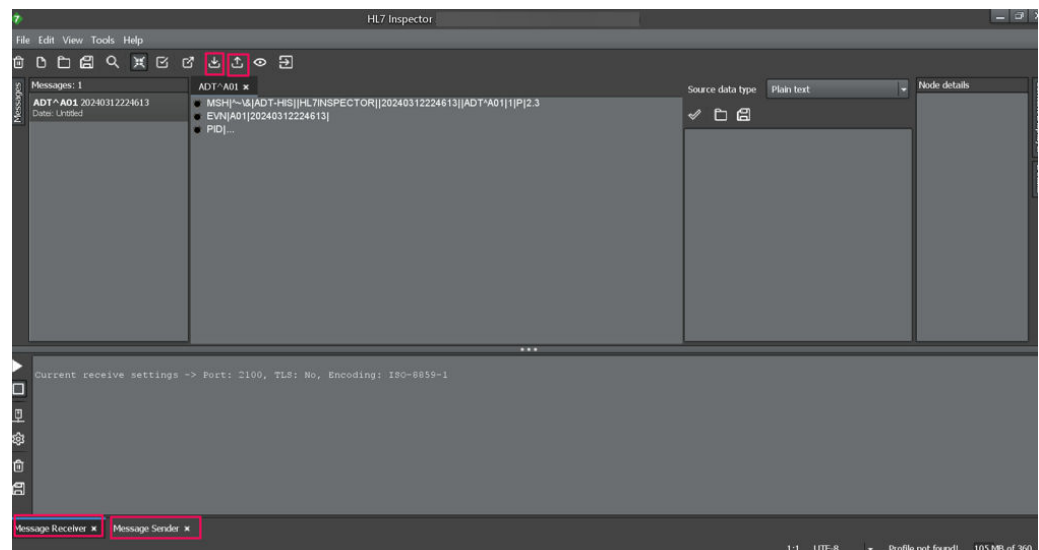
Download the connectivity agent installer to install the agent in your local environment. During installation, you associate the connectivity agent with the agent group identifier you generated when creating an agent group in Oracle Integration. Ensure that you restrict access to the folder in which you install the connectivity agent to only those users who need to stop and start the agent.

To download and install the connectivity agent, see *Agent Download and Installation in Using Integrations in Oracle Integration 3*.

3. Install and use the HL7 Simulator.

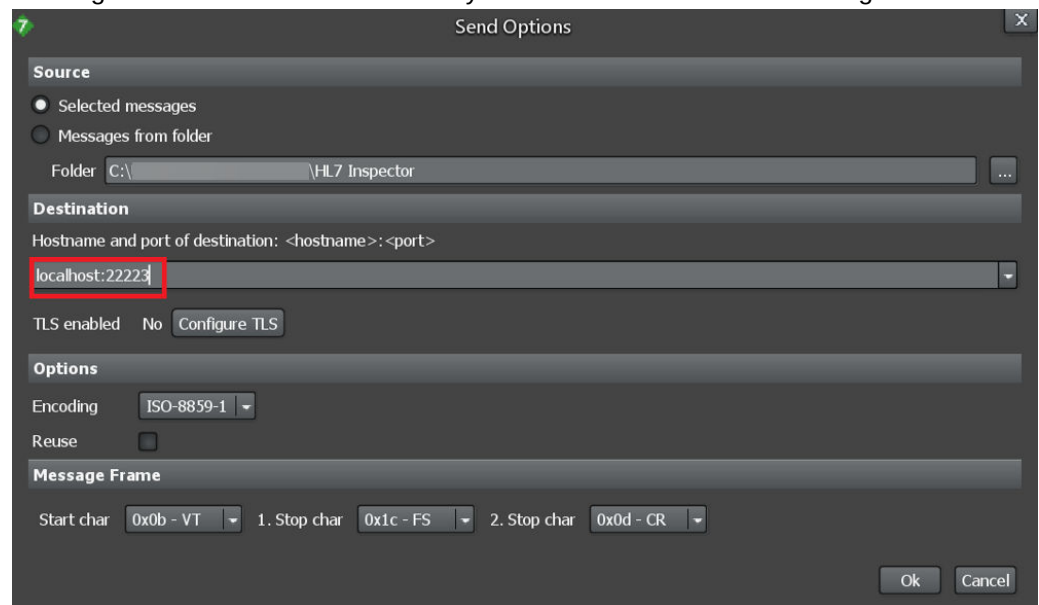
The HL7 Simulator is used to simulate the EMR application to send HL7 messages.

- [Download](#) and install the HL7 Simulator.
- Launch the HL7 Simulator application.
- From the top menu bar, select the Send/Receive icons. Message Receiver and Message Sender panes are visible at the bottom of the application window.



- d. In the Message Sender window, select the *Setup Send..* icon. In the <hostname>:<port> field enter localhost: 22223.

Message Sender simulates the EMR system to send HL7 V2.5.1 messages.



Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the MLLP Receiving Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter 22223 as the **Listener Port**.

3. Under **Access Type**, choose **Connectivity agent**, and select the agent group that you created earlier. See [Before You Install the Recipe](#).
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **Utility_Lookup** lookup table and update it as per your requirements.

1. In the Lookups section, click the lookup name.
2. In the **EmailTo** lookup key, enter an email ID of your choice to which you want to send the notification about the new appointment.
3. In the **EmailFrom** lookup key, enter an email ID of your choice from which you want to send the new appointment notification.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Launch an HL7 Inspector.
 - b. Send an HL7 2.5.1 message. For details, see step 3 in [Before You Install the Recipe](#).

Note

Ensure that the **Message Sender** is open, and Sender is sending to port 22223.

- c. In the HL7 Inspector, select **File** and then **File Open**. Browse to **SIU_S12_NewAppointment.hl7**. In the Import Options window leave the defaults and click **OK**.

Note

You can download the following sample hl7 file to run the recipe:
[NewAppointment.hl7](#).

- d. Click **Send Selected Message** in the Send Message window. This will send the selected HL7 SIU 2.5.1 message. You'll receive an acknowledgment message confirming that the HL7 message has been sent.

The **New Appointment Notification to Care Provider** integration flow is now successfully triggered, and it sends the new appointment details to the healthcare provider.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if a new appointment notification has been received in the email ID configured to receive the notification in the recipe's lookup table.

Related Documentation

- *Using the MLLP Adapter with Oracle Integration 3*
- [Live lab for Healthcare Use Case](#)

Send Notifications for Errors or Failures

Use this accelerator to send notification whenever an integration error or failure occurs.

Topics:

- [About This Accelerator](#)
- [Install the Accelerator](#)
- [Activate and Run the Accelerator](#)

About This Accelerator

This accelerator can be used in all your integrations to customize the way in which you receive error/failure 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.

To use the accelerator, you must install the accelerator and configure the connections and other resources within it. The accelerator supports the following type of notifications:

- **Email:** To get notifications in email, provide a valid email address to send the error details to.
- **PagerDuty:** To get notifications through PagerDuty, provide the ServiceID, EmailID, APIToken, and Assignee.
- **JIRA Ticket :** To get notifications in JIRA ticket, 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.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher

Install the Accelerator

On your Oracle Integration instance, install the accelerator to deploy and configure the integration and associated resources.

Install the accelerator on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the accelerator, configure the following resources:

Configure the Oracle Alerting Service Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, select **OAuth 2.0 Or Basic Authentication** as the security policy.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Configure the Oracle Alerting Service Invoke Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the connection URL as <code>https://www.oracle.com</code>
3. In the Security section, select **No Security Policy**.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

Edit the lookup tables of the accelerator and enter appropriate lookup key values.

The accelerator contains the following lookup tables:

- **ORCL-T-GENRIC_ENS_NOTIFICATION_DATA** stores the actual parameters for each notification type.
 - **ORCL-T-GENRIC_ENS_NOTIFICATION_SELECTOR** stores two parameters that define the notification type for the calling service.
 - **ORCL-T-GENRIC_ENS_NOTIFICATION_SELECTOR_MSGID** stores the parameters - Message_ID, Message_Subject, and Notification_Type.
1. In the Lookups section, click the lookup name.
 2. Enter appropriate values for the lookup keys in the lookup tables.
 - a. In the **ORCL-T-GENRIC_ENS_NOTIFICATION_DATA** lookup table, enter the notification type, and the values for the parameters in each notification type.

Notification_Type	Parameter	Value
Email	param1	Enter the email of the user to whom the notification will be sent.
	param2	Enter the email from which the email needs to 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	Specify a custom value
	param2	Specify a custom value

- b. In the **ORCL-T-GENRIC_ENS_NOTIFICATION_SELECTOR** lookup table, specify values for the following lookup keys:

Key	Value
Notification_Type	Specify the type of notification. For example: EMAIL, PAGERDUTY or JIRA.
IntegrationCode_Version	Specify the integration code and version of the calling service. For example: INT_CODE:VERSION(01.00.0000)

- c. In the **ORCL-T-GENRIC_ENS_NOTIFICATION_SELECTOR_MSGID** lookup table, specify values for the following lookup keys.

Key	Value
Message_ID	Specify the notification message's ID. For example, 4323
Message_Subject	Specify the notification message's subject. For example, Failure in DS.
Notification_Type	Specify the notification type. For example: EMAIL, PAGERDUTY or JIRA.

- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Activate and Run the Accelerator

After you've configured the connections and other resources, you can activate and run the accelerator.

- Activate the accelerator. See [Activate a Recipe or Accelerator](#).
- Run the accelerator.

To run the accelerator, you've to activate and run the integration in which it is used.

The integration which uses the accelerator must call the GenericErrorNotification service from the Global Fault Handler, and certain parameters have to be configured for the GenericErrorNotification service.

Configure the following parameters for the GenericErrorNotification service:

- **IntegrationCode_Version:** For example, 0RCL-BA-NS2SFDC_INVENTITEM_CREATE:01.00.0000.
 - **InstanceID:** Use the mapper function to get the instance ID of the calling service.
 - **FaultCode:** Map the errorCode from GlobalFaultObject.
 - **FaultReason:** Map the reason from GlobalFaultObject.
 - **FaultDetails:** Map the details from GlobalFaultObject.
3. If errors occur in the integration in which the accelerator is used, a notification will be sent through the specified notification type. For example, if you've configured the notification type as Email, then an email with the error details will be automatically sent to the specified email address.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*

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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 and configure the connections and other resources within the recipe.

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
- An account on Salesforce with the Administrator role

- Twilio
- An account on Twilio with the Administrator role

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.

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 the owner of a Salesforce lead is changed.

Log in to your Salesforce instance as an **Administrator** and complete the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.

- In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
- iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
- i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note it down.
Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.
 - c. On the New Outbound Message page:
 - i. Select **Lead** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter 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.

1. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the API key obtained earlier. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Twilio Connection

1. In the Connections section, click the connection name.
2. 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.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the properties of the integration flow with appropriate values.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - **NotificationMail**: This integration property holds the email address to which run-time 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.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Salesforce instance.

Note

If you're shown the Lightning 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 running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. 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 3](#)
- [Using the Salesforce Adapter with Oracle Integration 3](#)

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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe sends email notifications to order contacts when sales orders are fulfilled in Oracle ERP Cloud.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. Subsequently, you can activate the integration flow of the recipe. 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

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.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - 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.
6. In the Supported account types section, select who can use this application or access this API.
7. 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**.
 - iv. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Microsoft Email Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the Client ID obtained when you configured Microsoft Email. See Configure Microsoft Email .
Client Secret	Enter the Client Secret obtained when you configured Microsoft Email.
Scope	Enter the scopes URLs. You must include the <code>offline_access</code> scope along with the application-specific scopes. A blank space between each scope is also required. For example <code>https://graph.microsoft.com/Mail.ReadWrite</code> <code>https://graph.microsoft.com/Mail.Send</code> <code>offline_access</code>

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions** . . . on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **fromEmail**: This integration property holds the email address from which the order fulfillment notification emails are sent. Enter an email address of your choice.
 - b. **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 Team.
 - c. **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.
 - d. **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.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to the Oracle ERP Cloud instance as a Sales user with the Order Management role to fulfill an order.

Note

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.
 - i. 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.
 - ii. 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 running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. 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 3*
- *Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*

Send Salesforce Opportunity Documents to DocuSign

Use this recipe to send documents attached to a Salesforce opportunity to DocuSign.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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.

To use the recipe, you must install the recipe and configure the connections and other resources within it. 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.

① 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

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.

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.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user, for example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field.
Note down this user name.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce.

You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note down the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.
If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.
 - c. On the New Outbound Message page:

- i. Select **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_OPP0_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.
 - 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 **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](#).

Note

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.

7. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce.

Note

To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle DocuSign Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the 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 <code>signature extended</code> .
Instance Type	Select Sandbox .
Account ID	Enter the API account ID of your DocuSign account. See Configure DocuSign .

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**. You'll be redirected to the DocuSign login page.
6. Enter your DocuSign account credentials and click **Continue** to log in.
You'll be prompted to authorize the connection to the DocuSign application created previously.
7. 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.
8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
9. To return to the project workspace, click **Go back** .

Update Integration Properties

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **DocuSignAccountID**: This integration property holds the unique number associated with a DocuSign account. Enter the ID of your account. See [Configure DocuSign](#).
 - b. **NotificationMail**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page, click the **Opportunities** tab.

Note

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.
- d. Find a sample opportunity record that hasn't been closed, and click its name to open it.
- e. On the opportunity's page:
 - i. 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 running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
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.
 - b. 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 Salesforce Adapter with Oracle Integration 3](#)
- [Using the DocuSign Adapter with Oracle Integration 3](#)

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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)

- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 and configure the connections and other resources within the recipe.

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
- An account on Salesforce with the Administrator role
- SurveyMonkey
- An account on SurveyMonkey with the Administrator role

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.

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 complete the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
 - b. Click **New User** in the resulting page.
 - c. On the New User page:
 - i. Enter a first name and last name for the user. For example, **Integration User05**.
 - ii. In the **Email** field, enter a valid email address.

The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.

- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note it down.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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.
- ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.

- b. Click **New Outbound Message** in the resulting page.
- c. On the New Outbound Message page:
 - i. Select **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.

6. 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.
 - 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 resulting 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.
 - g. Get the SurveyMonkey API credentials.
 - i. Click **SETTINGS** to open the app's settings.
 - ii. Scroll down to the Credentials section, and note the values in the **Client ID** and **Secret** field.

Click the eye icon in the **Secret** field to view the value. You'll need the SurveyMonkey API credentials, that is the client ID and secret values while configuring the SurveyMonkey connection from Oracle Integration.
 - h. Select the scopes.
 - i. In the Scopes section, select the following scopes:
 - Create/Modify Surveys
 - Create/Modify Collectors
 - Create/Modify Responses
 - View Surveys
 - View Collectors
 - View Responses
 - View Response Details

- ii. Click **Update Scopes**.
You get a confirmation message that the scopes are successfully updated.
2. Create a survey in SurveyMonkey for which you want to receive responses.
 - a. Log in to your SurveyMonkey account at <https://www.surveymonkey.com/user/sign-in/>.
 - b. On the SurveyMonkey Dashboard, click **CREATE SURVEY** on the title bar.
 - c. Create a survey using one of the following options in the Create a new survey pane on the left:
 - Start from scratch
 - Copy a past survey
 - Buy Targeted Response
 - Import Questions
 - Start from template
 - Build it for me

Find information related to creating and managing surveys under the Design and Manage topic in the SurveyMonkey Help Center (<https://help.surveymonkey.com/>). Note down the name of the survey that you have created.
 - d. Click **Next** to preview your survey.
 - e. Click **Next** to view the COLLECT RESPONSES section.
 - f. Click **Send surveys your way**.

The COLLECT RESPONSES section displays the different ways in which you can collect the responses.
 - g. Click **Share a survey link**.

The COLLECT RESPONSES section displays the details of the web link collector. Note down the name of the collector. For example, Web Link 1.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the API key obtained earlier. 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.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the SurveyMonkey Connection

- In the Connections section, click the connection name.
- 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.

- Click **Save**. If prompted, click **Save** again.
- Click **Provide Consent**.
- 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.

- 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.

- 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.

- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

1. In the Integrations section, click **Actions** . . . on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **SurveyName**: This integration property holds the name of the survey that you have created in SurveyMonkey. To obtain this name, see [Configure SurveyMonkey](#).
 - b. **CollectorName**: This integration property holds the name of the web link collector that you have created in SurveyMonkey. To obtain this name, see [Configure SurveyMonkey](#).
 - c. **EmailNotification**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Salesforce instance.

Note

If you're shown the Lightning 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.

Note

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:
 - i. 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.

Note

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 running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
 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 3](#)
- [Using the Salesforce Adapter with Oracle Integration 3](#)

Send WhatsApp Notifications to Oracle ERP Cloud Lead Owners Using Twilio

Use this recipe to send a WhatsApp notification message to an Oracle ERP Cloud lead's new owner when the owner is changed.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe sends a WhatsApp notification message to the newly-assigned owner of an Oracle ERP Cloud sales lead using Twilio.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a lead's owner changes in Oracle ERP Cloud, the integration flow of the recipe is triggered. The recipe then receives the lead information and creates a notification message. Subsequently, Twilio is invoked by the recipe to send a notification message in WhatsApp to the newly-assigned lead owner's mobile number. Note

that for the recipe to work, you must ensure that a mobile number is associated with the newly assigned lead owner.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39447)
- Oracle ERP Cloud
- Twilio
- An account on Oracle ERP Cloud with the Administrator role
- An account on Twilio

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud and Twilio instances in order to successfully connect to these external systems using Oracle Integration and send messages to Oracle ERP Cloud users.

Configure Oracle ERP Cloud

To access Oracle ERP Cloud using Oracle Integration and retrieve information about Oracle ERP Cloud leads, you must perform certain configuration tasks on your Oracle ERP Cloud instance.

You'll require a separate user account on Oracle ERP Cloud to access it from Oracle Integration. Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks:

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Additionally, configure the following prerequisite tasks on your Oracle ERP Cloud instance.

1. Navigate to Application Composer.
2. Expand **Standard Objects** and then expand **Sales Lead**.
3. Click **Server Scripts**.
4. Under the **Triggers** tab, select **Action** and then select **Add for Field Triggers**.
5. In the **Trigger** field, select **Before Update in Database**.
6. In the **Trigger Name** field, enter Lead Assign.
7. In the Trigger Definition section, enter the following script in the expression text box:

```
if (isAttributeChanged('OwnerId')) {  
    return true;}

```

```
    else{  
    return false}
```

8. Click **Validate**.
You'll get a confirmation message that the script is parsed successfully.
9. Click **OK**.

Configure Twilio

To connect to Twilio from Oracle Integration and send messages through it, you must obtain the credentials associated with your Twilio account.

1. Log in to the Twilio console at <https://console.twilio.com/>.
2. Obtain the credentials associated with your Twilio account.
 - a. Click **Account** in the title bar, and select **General settings**.
 - b. Under **API Credentials**, 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
User Name	Enter the user name of the account created for Oracle Integration on the Oracle ERP Cloud instance. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle ERP Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Twilio Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
AccountSID	Enter the Account SID of the Twilio account. See Configure Twilio .
Auth Token	Enter the Auth Token of the Twilio account.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Set Up WhatsApp with Twilio

After you've configured the Twilio connection in Oracle Integration, you've to set up WhatsApp with Twilio in order to send WhatsApp notification messages with Twilio.

For details, see how to set up [WhatsApp Business API with Twilio](#).

While you wait for WhatsApp to formally approve your account, you can try [Twilio Sandbox for WhatsApp](#) to quickly test your recipe in a developer environment.

To send or receive WhatsApp messages to an end user from the Sandbox, you must first have any user join the Sandbox.

To join a Twilio Sandbox, send the message `join <your Sandbox keyword>` to your Sandbox number in WhatsApp.

- Twilio Sandbox is pre-provisioned with a Twilio phone number (For example, +1-415-523-8886). You can send the message to this number for testing.
- The Sandbox keyword is a unique string that you can get from the Twilio Console.

Update the Integration Property

Update the integration property of the integration flow with an appropriate value.

Note

This is an app-driven integration, and you can update the property value only before activating the recipe.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration property with appropriate value.

- **From:** This integration property holds the phone number to which the WhatsApp notification message is sent. Enter the mobile number of the newly-assigned owner of a Oracle ERP Cloud sales lead.

Note

For the purpose of testing, you can enter the free number that you received while joining a Twilio Sandbox. See [Set Up WhatsApp with Twilio](#).

3. Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle ERP Cloud instance with the Sales role.
 - b. Click **Sales** in the navigator.
 - c. Select and open an existing lead.

Optionally, click **Leads** and then click **+** to create a new lead. Enter the required details such as Lead Name, Contact Email, Owner, and so on, and then click **Save**.
 - d. Select a new owner for the lead from the drop-down list in the **Owner** field.
 - e. Click **Save**.

You've now successfully submitted the integration for execution.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. On the mobile phone of the newly-assigned lead owner, check for the WhatsApp notification message from the Twilio account. The message notifies the user about the Oracle ERP Cloud lead assignment, and also contains the lead ID using which the user can check details about the newly-assigned lead in Oracle ERP Cloud.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Twilio Adapter with Oracle Integration 3*

Share YouTube Video Posts on Socials

Use this recipe to connect to your YouTube channel, get the metadata of videos (such as title, description, and URL), and post the videos on LinkedIn and Twitter.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)

- [Activate and Run the Recipe](#)

About This Recipe

The recipe demonstrates how to connect to your YouTube channel, get all the playlists, fetch the metadata of videos (such as title, description, and URL), and post the videos to the OCI Streaming service as a JSON.

Two independent app-driven orchestration flows poll the OCI Streaming service, retrieve the JSON data, and share it on LinkedIn and Twitter, respectively.

Note

Each time the app-driven integrations poll the video post from OCI Streaming, a random video post is shared on the socials. A random integer is generated to obtain video postings from an array queried from the OCI Streaming service. For instance, if there are thirty videos in a playlist, a random number between 1 and 30 is generated using the JavaScript library, and based on that random number, the video post is fetched from the playlist array and posted on socials.

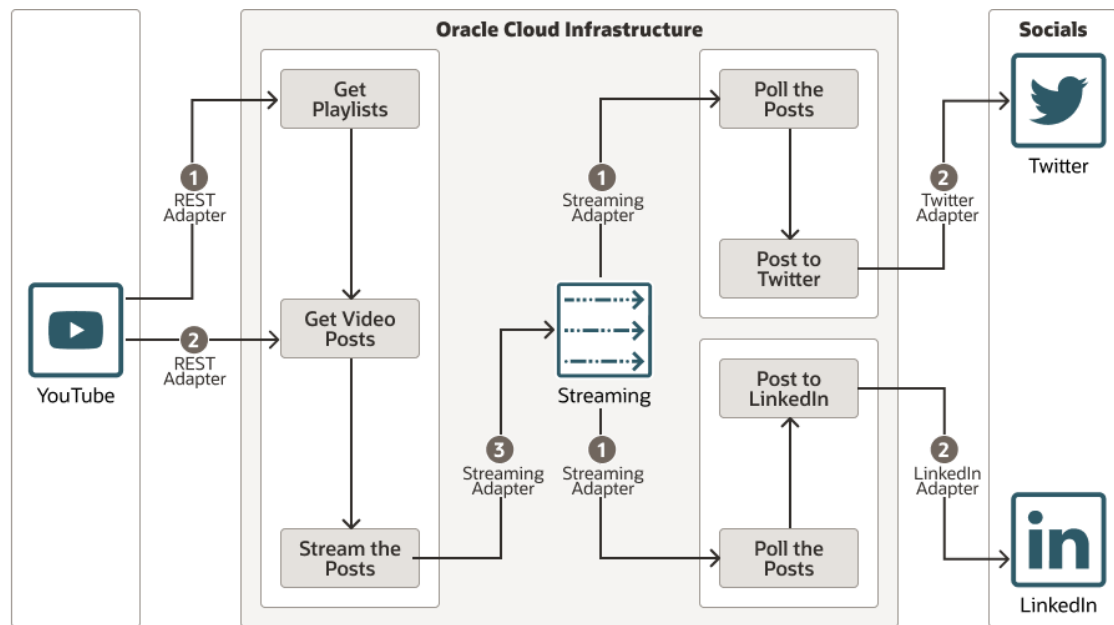
In this recipe, we use the OCI Streaming service to publish all the playlist video posts as a JSON, and two subscribers are connected to the same stream to consume the video posts.

System and Access Requirements

- OCI Streaming
- Compute Instance
- Oracle Integration

Recipe Schema

This section provides an architectural overview of the recipe.



This recipe has the following integrations:

- **YouTube OCI Streaming YT Posts:** This integration connects to your YouTube channel using REST APIs, gets all the playlists, fetches the metadata of videos (such as title, description, and URL), and posts videos to the OCI Streaming service as a JSON array.
- **Share YT Posts on Twitter:** This integration polls all the messages from the OCI Streaming service, and shares them on Twitter.
- **Share YT Posts on LinkedIn:** This integration polls all the messages from the OCI Streaming service, and shares them on LinkedIn.

The recipe has the following controls:

- **Exclude Playlist IDs:** Use this control to exclude playlists from being shared on socials. Enter comma-separated IDs of the playlists that you want to exclude in the `excludePlaylistIds` key of the lookup table. See [Configure the Lookup Table](#).
- **Don't share video posts on LinkedIn and Twitter:** Use this control to prevent sharing posts on socials. For the following keys, set value to **false** to prevent sharing videos on LinkedIn and Twitter, respectively.

- `publishVideoOnLinkedIn`
- `publishVideoOnTwitter`

The default value for both the keys is **true**.

- **Share number of video posts on LinkedIn and Twitter:** Use this control to specify the number of posts you want to share on LinkedIn and Twitter. You can configure this using the following keys in the lookup table.

- `publishVideoOnLinkedIn`
- `publishVideoOnTwitter`

The maximum possible value for the keys is 50, and the default value is 1.

Before You Install the Recipe

You must perform the following configuration tasks on your OCI, Twitter, and LinkedIn instances in order to successfully connect to these external systems using Oracle Integration and share posts on socials.

Create OCI Compute Instance

Create a compute instance to install the Oracle Integration agent. See [Create a Compute Instance](#).

Install Oracle Integration Agent on Compute Instance

Download and install the Oracle Integration agent in the compute instance you created.

Note

Before you install the Oracle Integration agent, ensure that you create an Agent group. Refer to the same group in the OCI Streaming connection.

See Download and Install the Connectivity Agent in *Using Integrations in Oracle Integration 3*.

Create OCI Stream

Create an OCI Stream named **YTPostStream**. The stream must have a single-partition which is used to retrieve video-related metadata as JSON. For this recipe, we have created a default stream pool as a public endpoint. See [Creating Streams](#).

Note

You can choose to create a new stream pool with a private endpoint. While doing so, ensure that the Oracle Integration agent is able to connect to your stream pool.

Configure a Twitter Application

Create a Twitter application to get access to the Twitter API, which is then used to configure the Twitter adapter. See Prerequisites for Creating a Connection in *Using the Twitter Adapter with Oracle Integration 3*.

Create a LinkedIn Application

Register an application in the LinkedIn developer console to obtain client ID and client secret, which is then used to configure the LinkedIn Adapter. See Prerequisites for Creating a Connection in *Using the LinkedIn Adapter with Oracle Integration 3*.

Create a Project in Google Cloud

Create a project in Google Cloud to connect to the YouTube APIs. See [Creating and managing projects](#).

After creating the project, you must:

- **Create API Key Credentials:** See [Create access credentials](#).
- **Enable YouTube Data API v3:** See [Enabling and Disabling Services](#).

You'll use this API key in the REST adapter configuration.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the YouTube REST Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select API Key Based Authentication .
API Key	Enter the Google API Key which you created in the Create a Project in Google Cloud section.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the Twitter Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Custom Security Policy selected.
Consumer Key	Enter the consumer key.
Consumer Secret	Enter the consumer secret.
Access Token	Enter the access token.
Access Secret	Enter the access token secret.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the LinkedIn Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the unique string provided to your client on the LinkedIn developer console.
Client Secret	Enter the client secret.
Scope	Enter the scope of the access request. Scopes enable you to specify which type of access you need, for example: r_basicprofile+r_emailaddress+w_share.

3. Click **Provide Consent**. Prompts LinkedIn to ask consent of the user in order to authorize the application's access to the LinkedIn profile.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table to maintain the configurable values.

1. In the Lookups section, click the lookup name.
2. Enter the following information in the lookup table.

Key	Description
channelId	Enter the ID of the YouTube channel from which you want to fetch video posts.
linkedinOwnerId	Enter the LinkedIn owner ID.
excludePlaylistIds	Enter comma-separated playlist IDs that you want to exclude.
excludePlaylistNamesForIdentification(Not in use)	Enter comma-separated playlist names that you want to exclude for reference. This variable is not used in any integration.
publishVideoOnLinkedIn	Set false if you don't want to share posts on LinkedIn. Default value is true .
publishVideoOnTwitter	Set false if you don't want to share posts on Twitter. Default value is true .
tags	Configure the tags that you want to attach with your Twitter posts.
noOfVideosToPublishOnLinkedIn	Update this number if you want to share more than one video on LinkedIn in a single post. Default value is 1. The maximum possible value is 50.
noOfVideosToPublishOnTwitter	Update this number if you want to share more than one video on Twitter in a single post. Default value is 1. The maximum possible value is 50.

Key	Description
emailTo	Enter the email ID on which you want to receive email notifications in case of any errors in the recipe.
emailFrom	Enter the email ID from which you want to send email notifications in case of any errors in the recipe.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Twitter and LinkedIn instances and check for the new posts.

Related Documentation

- [Using the Twitter Adapter with Oracle Integration 3](#)
- [Using the LinkedIn Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)

Sign and Verify Messages Using OCI Vault

Use this recipe to sign and verify messages using Oracle Cloud Infrastructure Vault (OCI Vault).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe enables you to sign and verify messages received from external systems using either Oracle or customer-managed keys in OCI Vault.

The recipe uses the OCI Vault REST APIs to sign and verify messages received from external applications. It contains two integration flows:

- **Sign Message:** Accepts message in string format as part of the request payload and invokes the [OCI Vault Sign REST API](#), which generates a message signature and sends the signed message as part of the response.
- **Verify Message:** Accepts signed message as part of the request payload and invokes the [OCI Vault Verify REST API](#), which verifies the message signature and sends the response as true or false.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- OCI Vault
- An account on OCI with the Administrator role
- OpenSSL version 3

Before You Install the Recipe

To access OCI Vault from Oracle Integration and sign or verify messages, you'll require to perform certain configuration tasks on OCI.

Log in to your OCI instance as an **Administrator** and perform the following tasks.

1. Create an OCI Vault. See [Managing Vaults](#).
2. Import an external private 2048-bit RSA key. See [Importing Asymmetric Keys](#).

When you create a new master encryption key or key version, you can import your own key material instead of letting the OCI Vault generate the key material internally. Importing your own key is called Bring Your Own Key (BYOK).

Note

This recipe uses BYOK, but you can also use Oracle managed and generated master encryption keys.

3. Get your Tenancy and User OCIDs. See [Where to Get the Tenancy's OCID and User's OCID](#).
4. Create your own private key or generate it from the Oracle Cloud Console. See [How to Generate an API Signing Key](#).
5. Get the key's fingerprint. See [How to Get the Key's Fingerprint](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the OCI Vault REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Enter the Cryptographic Endpoint URL of the Vault you created earlier. See Before You Install the Recipe .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Select OCI Signature Version 1 .
Tenancy OCID	Enter the Tenancy OCID. See Before You Install the Recipe .
User OCID	Enter the User OCID.
Private Key	Enter the API key generated.

Note

Before you upload the private key, you must convert it into the PKCS1 format.

Finger Print	Enter the key's fingerprint.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **Sign_Verify_Message_Lookup** lookup table and enter necessary values for the lookup keys.

1. In the Lookups section, click the lookup name.
2. Edit the lookup keys with appropriate values.

Key	Value
KeyID	Enter the OCID of the key. You can obtain the key OCID from the key details page.
KeyVersionId	Enter the OCID of the key version. You can obtain the key OCID from the version table.

Key	Value
SigningAlgorithm	Enter SHA_256_RSA_PKCS_PSS
EmailFrom	This is the email address from which notifications are sent in case of errors. Leave the default email address no-reply@oracle.com or enter an email of your choice.
EmailTo	Enter an email to which notifications are to be sent in case of errors.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe from an external application.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
 - d. To sign a message from an external application, send a POST request to this endpoint URL along with the message to be signed. Provide the message in the POST request's **Body**. See the subsequent step for example request payload.
The recipe generates a signature for the message you pass in the request body.
 - e. To verify a message and signature from an external application, send a POST request to this endpoint URL along with the message and signature to be verified. Provide the message and signature in the POST request's **Body**. See the subsequent step for example request payload.
The recipe verifies the signature against the message you pass in the request body.

3. Test the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Sign Message** integration flow, then select **Run**.
 - b. On the Configure and run page, in the Request section, click the **Body** tab and enter the message as request payload.

Example request payload to sign message:

```
{
  "message ":
  " {\\"EmpId\\":\\"011\\",\\"Name\\":\\"xyz\\",\\"Email\\":\\"xyz@oracle.com\\"}"
}
```

- c. In the Response section of the Configure and run page, you'll find the response returned. The response will be the signature of the message.
- d. In the Integrations section of the project workspace, click **Actions**  on the **Verify Message** integration flow, then select **Run**.

- e. On the Configure and run page, in the Request section, click the **Body** tab and enter the message and its signature as request payload.

Example request payload to verify message:

```
{
  "signature":
    "wX4vpomM4aX0djjRHvCKN30xkx0ggjixUSXVkpGCU2IijhI6ElUda9h0P5RynDF7nJrDd8D
    VKpFrdti9jT68APD4UTyqS+Dt7M/
    qsGzyjqd0sAEMeLLhYrZVMD+F3m49w1YGTaase7o7PtSVW7uhoJ2Mt9i/
    A4fYVtYoR0T3IXK4DBr6==",
  "message ":
    " {\"EmpId\": \"011\", \"Name\": \"xyz\", \"Email\": \"xyz@oracle.com\"}"
}
```

- f. In the Response section of the Configure and run page, you'll find the response returned. The response would be either true or false depending on the validity of the signature against the message.

For example:

```
{
  "isSignatureValid" : "true"
}
```

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

Related Documentation

- [Using the REST Adapter with Oracle Integration 3](#)

Shortlist Resumes Automatically Using Slack and OpenAI

Use this recipe to automatically evaluate resumes posted in a Slack channel against a specific job description and notify the channel if a resume is shortlisted.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe leverages OpenAI to process resumes that are posted to a Slack channel for a particular job description and notifies the channel if a resume is shortlisted.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a resume is posted in a Slack channel for a particular job description, the integration flow of the recipe gets triggered. The recipe then uses OpenAI to process and evaluate if the resume can be shortlisted. If the resume is shortlisted, a notification is sent to the same Slack channel.

System and Access Requirements

- Oracle Integration
- OpenAI
- Slack

Before You Install the Recipe

You must perform the following configuration tasks in Slack to create a workspace, App, API credentials, and set permissions.

Configure Slack

1. Create a Slack workspace

- Open the Slack get started page using the following URL: <https://slack.com/get-started#/createnew>.
- Enter your email address and click **Continue**.
- Enter the confirmation code that you receive in your email, click **Create a workspace**, and follow the prompts.

2. Create a Slack App

- Log in to the Slack app platform at <https://api.slack.com>.
 - In the Slack API title bar, click **Your Apps**.
 - Click **Create an App**, and in the resulting Create an app dialog, select **From scratch**.
 - In the Name app & choose workspace dialog, enter the following details:
 - In the **App Name** field, enter the name of your app. For example, Oracle Integration App.
 - In the **Pick a workspace to develop your app in** field, select your workspace from the drop-down list.
 - Click **Create App**.
- The app gets created and its Basic Information page appears.

3. Get the Slack App API credentials

- In the 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 details are required to configure connections to your Slack instance from Oracle Integration.

4. Set permissions for your Slack App

- In the left menu, under **Features**, select **OAuth & Permissions**.
 - In the OAuth & Permissions page, go to the Redirect URLs section.
 - 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:write
 - groups:write
 - im:write
 - mpim:write

Note the scope details as they are required to configure connections to your Slack instance from Oracle Integration.
- g. Press **Enter**.

5. Configure event subscriptions for your Slack App

- a. In the left menu, under **Features**, select **Event Subscriptions**.
- b. In the **Request URL** field, enter the Oracle Integration endpoint URL.
For example: `https://<username>:<password>@<oic host>/ic/api/integration/v2/flows/rest/project/ORCL-R-SLACK_OPENAI_RESUME_SL/SLACKEVENTSUBSCRIPTION/1.0/eventSub`
- c. Expand the **Subscribe to bot events** section, click **Add Bot User Event** and configure the Event Name, Description, and Required Scope fields.

Event Name	Description	Required Scope
file_shared	A file was shared	files:read
message.channels	A message was posted to a channel	channels:history
message.groups	A message was posted to a private channel	groups:history
message.im	A message was posted in a direct message channel	im:history

- d. Expand the **Subscribe to events on behalf of users** section, click **Add Workspace Event** and configure the Event Name, Description, and Required Scope fields.

Event Name	Description	Required Scope
file_shared	A file was shared	files:read
message.channels	A message was posted to a channel	channels:history
message.im	A message was posted in a direct message channel	im:history
message.mpim	A message was posted in a multiparty direct message channel	mpim:history

- e. Click **Save Changes**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Before You Install the Recipe .
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter channels:write groups:write im:write mpim:write.

3. Click **Provide Consent**.
A new browser window opens to approve access to Slack.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Slack REST Connection

1. In the Connections section, click the connection name.
2. Enter the url `https://slack.com`.
3. In the Security section, select **No security policy**.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. Click **Save**. If prompted, click **Save** again.
6. To return to the project workspace, click **Go back** .

Configure the OpenAI Connection

1. In the Connections section, click the connection name.
2. Enter the API key.
3. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

The recipe contains the following three lookup tables that you can edit as necessary. Edit the lookup tables to enter appropriate values for the lookup keys.

- **EMAIL_PROPERTIES**: Use this lookup to configure the email fields.

Key	Value
from	Enter the email address from which error notifications have to be sent.
to	Enter the email address to which error notifications have to be sent.
subject	Enter the email subject for the error notifications.

- **SLACK_PROPERTIES**: Use this lookup to configure the Slack authorization type and channel.

Key	Value
Authorization	In the Slack app, go to Features, OAuth & Permissions, OAuth Tokens and copy the required token, either User OAuth Token or Bot User OAuth Token .
channel	Name of the Slack channel you have created for shortlisting resumes using this recipe.

- **OPENAI_PROPERTIES**: Use this lookup to configure the role.

Key	Description/Value
Role	Enter the user role name. This recipe has the following preconfigured keys and their values: – Role_1: developer – Role_1_prompt: You are an IT Talent Acquisition Assistant. Your job is to evaluate candidate resumes based on the provided Job Description and recommend whether the candidate should be shortlisted or not. Provide a brief rationale for each recommendation. – Role_2: user

1. In the Lookups section, click the lookup name.

2. Edit the lookup table as necessary. In the respective lookup table, map the values for the “Key” with respective prompts in the “Value”.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - Enter the job description and resume summary in the specific Slack channel as shown in the following example.

For example:

JD: Job Title: Senior Java Developer Location: Remote (Preference for candidates in North America) Required Skills:

- 6+ years of hands-on Java development experience
- Strong knowledge of Spring Boot and Spring Cloud
- Expertise in designing RESTful APIs
- Experience working with AWS services: EC2, Lambda, SQS, DynamoDB
- Solid understanding of microservices architecture
- CI/CD exposure (Jenkins, GitHub Actions)
- Excellent problem-solving and communication skills

Nice-to-Have:

- Knowledge of Kubernetes or Docker
- Experience working in Agile/Scrum teams
- Prior work in fintech or banking domain

Education: Bachelor's degree in Computer Science or related field

Resume Summary: 6 years Java experience using Spring Boot to develop REST APIs. Built microservices for e-commerce platforms. Familiar with AWS EC2 and DynamoDB, but limited exposure to Lambda and SQS. Worked in Agile teams and basic CI/CD with GitLab. No fintech/banking experience.

Education: B.Sc. in Computer Engineering.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify that a notification is sent to the Slack channel indicating whether the resume has been shortlisted or not.

Related Documentation

- *Using the Slack Adapter with Oracle Integration 3*

Store and Manage Objects in Amazon S3

Use this recipe to store and manage objects in an Amazon S3 bucket.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to store and manage objects in an Amazon S3 bucket.

To use the recipe, you must install the recipe and configure the connection and other resources within it. You can perform the following operations using this recipe:

- Upload an object into an Amazon S3 bucket
- List objects in an Amazon S3 bucket
- Delete an object in an Amazon S3 bucket
- Download an object from an Amazon S3 bucket

Before You Install the Recipe

To access Amazon Web Services using Oracle Integration and successfully store and manage objects in an Amazon S3 bucket, you have to perform certain configuration tasks.

Configure Amazon S3

Complete the following configuration tasks to successfully connect to AWS from Oracle Integration and store and manage objects in an Amazon S3 bucket.

1. 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 to store and manage objects.
 - 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.
 - i. Scroll to the Buckets section. All available buckets are listed under it.
 - ii. Select a bucket.
 - d. Alternatively, create and choose a new bucket.
 - i. Click **Create bucket**.
 - ii. In the Create bucket dialog, enter a unique name for the bucket, for example oracle-bucket.
 - iii. In the **AWS Region** field, select a region from the drop-down list
 - iv. Click **Create bucket**. The new bucket is listed under the Buckets section.
 - v. Select the bucket.
4. Note the **AWS Region** of the bucket. You will need this information when you configure the Amazon S3 connection from Oracle Integration.
 - a. Select the bucket.
 - b. 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Amazon S3 Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the AWS region where you have created the Amazon S3 bucket. For example, us-east-1.
3. In the Security section, enter the following details:

Field	Information to Enter
Access Key	Enter the access key that you obtained. See Configure Amazon S3 .
Secret Key	Enter the secret key that you obtained.
AWS Region	Choose the AWS region where you have created the Amazon S3 bucket. For example, US East (N. Virginia) Region (us-east-1).

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Test the recipe in Oracle Integration.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle AWS S3 Handler** integration flow, then select **Run**.
 - c. Test the object upload, list, delete and download operations.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify the results of each operation (upload, list, delete, download).
 - a. Log in to your AWS Management Console at <https://console.aws.amazon.com/>.
 - b. Navigate to the Amazon S3 page.
 - c. Select the Amazon S3 bucket from the Buckets section of the page.
 - d. Verify that the operations you performed while testing the recipe are successful.

Related Documentation

- [Using the Amazon Simple Storage Service \(S3\) Adapter with Oracle Integration 3](#)

Sync Accounts in Salesforce with Zendesk Organizations

Use this recipe to synchronize accounts in Salesforce with Zendesk organizations.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is a bidirectional recipe. It creates a corresponding organization record in Zendesk when an account record is created in Salesforce. In addition, it creates corresponding account records in Salesforce for Zendesk organizations.

To use this recipe, you must install the recipe and configure the connections and other resources within it.

When an account is created in Salesforce, the app-driven integration flow (**Oracle Salesforce Zendesk Acc2Org Sync**) of the recipe is triggered, and it creates a corresponding organization record in Zendesk. Further, you can run the scheduled integration flow (**Oracle Zendesk Salesforce Organization2Acc Sync**) of the recipe manually or specify an execution schedule for it. When triggered, this integration flow reads the organization records from the Zendesk instance and, in turn, creates corresponding accounts in your Salesforce instance.

Basic data, such as organization ID, name, domain name, account name, account description, and website, are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Zendesk
- An account on Zendesk with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Zendesk instances in order to successfully connect to these external systems using Oracle Integration and synchronize records between them.

Configure Salesforce

To access Salesforce from Oracle Integration and create/read account records, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the **Oracle Salesforce Zendesk Acc2Org Sync** integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce account is created. Finally, create a custom field for Salesforce account records to store Zendesk organization IDs.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

Log in to your Salesforce instance as an Administrator and perform the following tasks:

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. In the **Existing Profile** field, select **Standard User**.
 - ii. Enter a name for the new profile (for example, API Enabled), and click **Save**.

The new profile is now saved, and the Profile Detail page of the new profile is displayed.

- d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the **Administrative Permissions** section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the **Standard Object Permissions** section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note it.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the **Quick Links** section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.

A new security token is sent to the email address associated with the account. Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
5. On your Salesforce instance, create an outbound message to trigger the **Oracle Salesforce Zendesk Acc2Org Sync** integration flow of this recipe.

On the Salesforce Setup page:

- a. Select **Account** in the **Object** field, and click **Next**.
- b. Enter the following details for the new outbound message:
 - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the message.
 - In the **Endpoint URL** field, enter the following URL:

```
https://<OIC_FQDN:OIC_SSL_PORT>/ic/ws/integration/v1/flows/  
salesforce/ORCL-R-SFDC_ACC_SYN_TO_ZEN_ORG/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 **Account 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 account is created.

- 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 **Account** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:
 - Enter the rule's name.
 - Optionally, enter a description for the rule.
 - In the **Evaluation Criteria** section, select the **created** radio button.
 - In the **Rule Criteria** section, select the **formula evaluates to true** option, and define the following rule criteria:

 (Account: Account Name NOT EQUAL TO null) AND (Account: Description NOT EQUAL TO null) AND (Account: Zendesk_Org_Id EQUALS null)
 - Click **Save & Next**.
 - iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
 - iv. On the Select Existing Actions page:
 - Select **Outbound Message** in the **Search** field.
 - In the **Available Actions** box, select the outbound message you created previously. For example, Outbound Message: *name of your outbound message*.
 - Click **Add** to move your selection to the **Selected Actions** box.
 - Click **Save**.
 - v. Click **Done**.

The new workflow rule is created and displayed on your browser.

- d. Click **Activate** to activate the rule.
7. Create a custom field for account records.

This recipe uses unique IDs associated with organization records in Zendesk to synchronize Zendesk organizations with Salesforce accounts. Create a custom field for Salesforce account records to hold the Zendesk IDs.

- a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Customize**, then **Accounts**, and then click **Fields**.
- b. On the Account Fields page, scroll to the **Account Custom Fields & Relationships** section and click **New**.
- c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.

- ii. Enter the following details for the new custom field:
 - Enter `Zendesk_Org_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 Account Fields page, you can see the new field added under the **Account Custom Fields & Relationships** section.

Configure Zendesk

To access Zendesk from Oracle Integration and create or read organization records, you'll need to create an OAuth client on your Zendesk instance. See [Create an OAuth Client in Using the Zendesk Adapter with Oracle Integration 3](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Zendesk Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Zendesk host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID of the OAuth application created in Zendesk. See Configure Zendesk .
Client Secret	Enter the client secret of the OAuth application created in Zendesk.
Scope	Enter write.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**. You'll be redirected to the Zendesk login page.
7. Enter your Zendesk account credentials and click **Sign in**.
8. After you see an access allowed message, switch back to the Zendesk Connection window of Oracle Integration to test your connection.
9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
10. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration property for the **Oracle Salesforce Zendesk Acc2Org Sync** integration flow.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo**: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Configure the Integration Flow

Edit the **Oracle Zendesk Salesforce Organization2Acc Sync** integration flow, and define the mapping for the **Map to createAccountInSalesforce** element.

1. In the Integrations section, click the integration flow name.
2. On the integration canvas, click the **Map to createAccountInSalesforce** element, and then click **Actions** and select **Edit** to open the mapper.
3. In the Target pane, find the **External ID Field Name** element. Click the element to open the expression editor at the bottom.
4. In the editor, click the **Switch to Developer View** icon, and enter the following expression: "Zendesk_Org_Id__c".
5. Click **Save**, then **Validate**.
6. Click **Close** to return to the integration canvas.
7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe to create Zendesk organizations for new Salesforce account records.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page (in the Classic UI), click the **Accounts** tab.
 - c. On the Accounts Home page, click **New**.
 - d. On the New Account page, enter the account details, such as name, description, and so on, and ensure that the Zendesk_Org_Id field is blank. Click **Save**.

You've now successfully triggered the recipe, which creates a corresponding organization record in Zendesk.

3. Run the recipe to create accounts in Salesforce for Zendesk organization records.
 - a. In the Integrations section of the project workspace, click **Actions** on the integration flow (**Oracle Zendesk Salesforce Organization2Acc Sync**), then select **Run**.

The Configure and run page is displayed, where you can specify a value for the LastRunTime parameter. This parameter stores the date and time of the most recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Zendesk organization records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the organization records from the beginning, set the value as zero.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Zendesk instance and check for the new organizations created.
 - a. On the Zendesk home page, click **Organizations** in the left navigation pane.
 - b. Search for the required organization records on the Organizations page.
6. Log in to your Salesforce instance and check for the new accounts created.
 - a. On the Salesforce Setup page (in the Classic UI), click the **Accounts** tab.
 - b. On the Accounts Home page, select **All Accounts** in the **View** field. Click **Go!** if necessary.
The list of accounts is displayed.
 - c. To view an account record, click its name.

Related Documentation

- [Using the Salesforce Adapter with Oracle Integration 3](#)
- [Using the Zendesk Adapter with Oracle Integration 3](#)

Sync Contacts Between Salesforce and Oracle B2C Service

Use this recipe to synchronize contact records between Salesforce and Oracle B2C Service.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates a corresponding contact in Oracle B2C Service when a new contact record is created in Salesforce. After successful creation of contact, the Oracle B2C Service contact Id will be saved in the custom field of the Salesforce contact.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a Salesforce contact is created, the integration flow of the recipe is triggered via platform events configured in Salesforce, and it creates a corresponding contact record in Oracle B2C Service. Basic data such as first name, last name, address, and so on, are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Salesforce
- Oracle B2C Service

- An account on Salesforce with the Administrator role
- An account on Oracle B2C Service with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Oracle B2C Service instances in order to successfully connect to these external systems using Oracle Integration and synchronize contact records between Salesforce and Oracle B2C Service.

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve data about Salesforce contacts that are created or updated, you must perform certain configurations 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.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.

- In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
- iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
- i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.
Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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. See Find Your Current Salesforce API Version.
4. Identify your Salesforce instance type. See Identify Instance Type of Your Salesforce Organization.
5. Create a custom field to store the Oracle B2C Service contact id.
 - a. On the Salesforce Setup page:
 - i. Scroll to Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Contacts**, and then click **Fields**.
 - b. On the Contact Fields page, scroll to the Contact Custom Fields and Relationships section and click **New**.
 - c. 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, RNContact_Id.
Length	Enter the maximum length for the text field. For example, 255.
Field Name	This field is automatically populated.
Description	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**.
6. Configure the Salesforce platform event. The platform event notifies external systems about changes to the Salesforce contact records.
 - 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 a label for the field. For example, ContactCreate_PE.
Plural	Enter the plural form of the label field.
Object Name	This field is automatically populated.
Description	If required, enter a description for the new platform event.

Field	Information to Enter
Publish Behavior	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**.
7. Create custom fields to store Oracle B2C Service contact record details.
 - a. In the Custom Fields & Relationships section of the Platform Event page, click **New**.
 - b. On the resultant New Custom Field page, select the **Text** radio button, and click **Next**.
 - c. Enter the following details for the custom field:

Field	Information to Enter
Field Label	Enter the field name. For example, ContactId_PE.
Field Name	This field is automatically populated.
Length	Enter the maximum length of the text field. For example, 255.
Description	Enter a description for the custom field.

- d. Click **Save**.
 - e. Create as many custom fields as required by repeating steps a-d. For example, you can create custom fields such as FirstName_PE, LastName_PE, MailingCity_PE, PostalCode_PE and so on.
8. Create and activate a process that triggers the platform event.
 - a. In 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.
API Name	This field is automatically populated.
Description	If required, enter a description for the process.
The process starts when	Select A record changes .

- iii. Click **Save**.
- c. Configure the process trigger.

- i. Click the **Add Object** 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 Contact .
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 integration user from triggering the process.

Field	Information to Enter
Field	Search for <code>Contact.FirstName</code> and select it.
Operator	Select Is null .
Type	Select Boolean .
Value	Select False .

- iv. Create another condition to restrict the integration user from triggering the process.

Field	Information to Enter
Field	Search for <code>Contact.LastName</code> and select it.
Operator	Select Is null .
Type	Select Boolean .
Value	Select False .

- 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**.
- e. Add action to the process.
 - i. In the **Immediate Actions** box beside the **TRUE** arrow, click **Add Action**. Enter a name for the action. For example, `CreateContact_Trigger`.
 - ii. In the Create a Record pane, under **Record Type**, select the Platform event that you created earlier in step 6.
 - iii. In the Set Field Values section, create rows for all the custom fields that you created earlier in step 7.
To create a row for a custom field:
 - Under **Fields**, type the name of the custom field.
 - Under **Type**, select **Field Reference**.

- Under **Value**, select the corresponding value of the custom field.
Repeat the above steps to create rows for all the custom fields.
- 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**.

Configure Oracle B2C Service

To access Oracle B2C Service using Oracle Integration and create corresponding contact records for Salesforce contacts, you must log in to your Oracle B2C Service as an **Administrator** and complete certain prerequisite tasks. See Prerequisites for Creating a Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle B2C Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the WSDL to use. See Prerequisites for Creating a Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.
3. In the Security section, enter the username and password that you obtained.
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must create a contact in Salesforce.

- a. Log in to your Salesforce instance.
- b. On the Salesforce Setup page, click the **Contacts** tab.
- c. On the Contacts Home page, click **New**.
- d. On the New Contact page, enter the contact details such as first name, last name, address, and so on, and then click **Save**.

You've now successfully triggered the recipe by creating a contact in Salesforce.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check for the new contact created in Oracle B2C Service.
 - a. Log in to your Oracle B2C Service instance.
 - b. Search for the new contact created.
 - c. Open the contact and check if the basic data matches that of the new contact created in Salesforce, and the Oracle B2C Service contact id got updated in the custom field (RNContact_Id).

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*

Sync Contacts Between SurveyMonkey and Oracle NetSuite

Use this recipe to synchronize contacts that you add in SurveyMonkey to Oracle NetSuite.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes contacts that you add in a SurveyMonkey group to a customer's contact list in Oracle NetSuite. The name of the SurveyMonkey group must match the name of the customer in Oracle NetSuite for the recipe to work.

To use the recipe, you must install the recipe and configure the connections within it. When a contact is added to a SurveyMonkey group, the **Oracle SurveyMonkey NetSuite Customer Contact Sync** integration flow is triggered. The recipe then adds the contact to a customer's contact list in Oracle NetSuite.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- SurveyMonkey
- An account on SurveyMonkey with the Administrator role
- Oracle NetSuite, Version 2019.1.0
- An account on Oracle NetSuite with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your SurveyMonkey instance and Oracle NetSuite instance in order to successfully connect to these external systems using Oracle Integration and synchronize contacts.

Configure SurveyMonkey

You must perform the following prerequisite tasks in your SurveyMonkey instance.

1. Create and configure a SurveyMonkey app.
 - a. Log in to the SurveyMonkey Developer console at <https://developer.surveymonkey.com/>.
 - b. Click **My Apps** on the title bar of the SurveyMonkey Developer home page.
 - c. On the My Apps page, click **Add a New App**.
 - d. In the resulting APP CREATION dialog, enter the app's name in the **App Nickname** field, and select **Private App** as the **App Type** option.
 - e. Click **Create App**.

SurveyMonkey creates an app and displays the client ID and secret details on the resulting 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.
2. 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.

3. Select the scopes.
 - a. In the Scopes section, select the following scopes:
 - Create/Modify Contacts
 - View Contacts
 - View Users
 - Modify Groups
 - View Group
 - Create/Modify Workgroups
 - Create/Modify Workgroup Members
 - Create/Modify Workgroups Shared Resources
 - b. Click **Update Scopes**.

You get a confirmation message that the scopes are successfully updated.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See [Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).
2. Perform the TBA-related configuration tasks. See [Prerequisites for the Token-Based Authentication Security Policy in *Using the Oracle NetSuite Adapter with Oracle Integration 3*](#).

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Lists	Customers	Full
Lists	Contacts	Full
Lists	Employee	Full
Lists	Locations	Full

Tab Name	Permission	Level
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the SurveyMonkey Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID of the OAuth application created in SurveyMonkey. See Configure SurveyMonkey .
Client Secret	Enter the client secret of the OAuth application created in SurveyMonkey.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Provide Consent**.
5. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

You'll now be redirected to the SurveyMonkey login page.

6. Enter your SurveyMonkey account credentials and click **Log In**.

You are now displayed a page that prompts you to authorize the connection to the SurveyMonkey application created previously.

7. On the SurveyMonkey web page, click **Authorize**.

You're informed that access to SurveyMonkey is allowed. You can now switch back to the SurveyMonkey Connection window of Oracle Integration to test your connection.

8. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

9. To return to the project workspace, click **Go back** .

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle NetSuite WSDL URL. For example, `https://webservices.netsuite.com/wsd/NetSuite_application_version/netsuite.wsdl`,

where <NetSuite_application_version> is the version of the NetSuite application. See [Configure Oracle NetSuite](#).

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See <i>Create an Integration Record for Oracle Integration in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See <i>Create an Access Token for the User Account in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier. See <i>Make a Note of the NetSuite Account ID in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Ensure that a new contact is present in SurveyMonkey. If required, add a new contact with contact information. See [Using Custom Data to Store Contact Info](#) in the SurveyMonkey Help Center.
3. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Check if a new customer record with contact details (corresponding to the new contact in SurveyMonkey) has been created in Oracle NetSuite.

- a. Log in to your Oracle NetSuite instance.
- b. In the global search bar on top of the Oracle NetSuite home page, enter the customer name or related keywords and press Enter.
- c. In the search results, locate the new customer record and view its details. Basic contact information such as name, email, and phone number must be displayed.

Related Documentation

- *Using the SurveyMonkey Adapter with Oracle Integration 3*
- *Using the Oracle NetSuite Adapter with Oracle Integration 3*

Sync Contacts in Salesforce with Zendesk User Accounts

Use this recipe to synchronize contact records in Salesforce with Zendesk user records.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is a bidirectional recipe. It creates a corresponding user record in Zendesk when a contact record is created in Salesforce. In addition, it creates corresponding contact records in Salesforce for Zendesk user records.

To use this recipe, you must install the recipe and configure the connections and other resources within it.

When a contact is created in Salesforce, the app-driven integration flow (**Oracle Salesforce Zendesk Contact2User Sync**) of the recipe is triggered, and it creates a corresponding user record in Zendesk. Further, you can run the scheduled integration flow (**Oracle Zendesk Salesforce User2Contact Sync**) of the recipe manually or specify an execution schedule for it. When triggered, this integration flow reads the user records from the Zendesk instance and, in turn, creates corresponding contact records in your Salesforce instance. Basic data, such as name, email, phone, user ID, and so on, are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Zendesk
- An account on Zendesk with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Zendesk instances in order to successfully connect to these external systems using Oracle Integration and synchronize records between them.

Configure Salesforce

To access Salesforce from Oracle Integration and create/read contact records, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create a custom field for Salesforce contact records to store Zendesk user IDs, and configure the generation of change-event notifications for the Contact entity.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

Log in to your Salesforce instance as an Administrator and perform the following tasks:

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. In the **Existing Profile** field, select **Standard User**.
 - ii. Enter a name for the new profile (for example, API Enabled), and click **Save**.
The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the **Administrative Permissions** section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the **Standard Object Permissions** section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.

- In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
- iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
- i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note it.
Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
- i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the **Quick Links** section, click **Edit my personal information**.

- iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account.
Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
5. Create a custom field for contact records.

This recipe uses unique IDs associated with user records in Zendesk to synchronize Zendesk user accounts with contacts in Salesforce. Create a custom field for Salesforce contact records to hold the Zendesk IDs.

- a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Customize**, then **Contacts**, and then click **Fields**.
 - b. On the Contact Fields page, scroll to the **Contact Custom Fields & Relationships** section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter `Zendesk_User_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 Contact Fields page, you can see the new field added under the **Contact Custom Fields & Relationships** section.

6. Configure the generation of change-event notifications for the Contact entity.
This configuration is required for the successful execution of the app-driven integration flow (**Oracle Salesforce Zendesk Contact2User Sync**) of the recipe. With this configuration, when a new contact is created in Salesforce, a change-event notification is generated to trigger the app-driven integration flow of the recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Develop**, then click **Change Data Capture**.

- b. On the Change Data Capture page, from the list of available entities, select **Contact**, and then click **Add**.
- c. Click **Save**.

Configure Zendesk

To access Zendesk from Oracle Integration and create or read user records, you'll need to create an OAuth client on your Zendesk instance. See *Create an OAuth Client in Using the Zendesk Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce. Note: To the password, you must also append the security token generated for the same account.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Zendesk Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Zendesk host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID of the OAuth application created in Zendesk. See Configure Zendesk .
Client Secret	Enter the client secret of the OAuth application created in Zendesk.
Scope	Enter <code>write</code> .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**. You'll be redirected to the Zendesk login page.
7. Enter your Zendesk account credentials and click **Sign in**.
8. After you see an access allowed message, switch back to the Zendesk Connection window of Oracle Integration to test your connection.
9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
10. To return to the project workspace, click **Go back** .

Update Integration Property

Update the integration property for the **Oracle Salesforce Zendesk Contact2User Sync** integration flow.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo:** This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Configure the Integration Flow

Edit the **Oracle Zendesk Salesforce User2Contact Sync** integration flow, and define the mapping for the **Map to createContactInSalesforce** element.

1. In the Integrations section, click the integration flow name.
2. On the integration canvas, click the **Map to createContactInSalesforce** element, and then click **Actions**  and select **Edit** to open the mapper.
3. In the Target pane, find the **External ID Field Name** element. Click the element to open the expression editor at the bottom.

4. In the editor, click the **Switch to Developer View** icon, and enter the following expression: "Zendesk_User_Id__c".
5. Click **Save**, then **Validate**.
6. Click **Close** to return to the integration canvas.
7. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe to create Zendesk organizations for new Salesforce contact records.

- a. Log in to your Salesforce instance.
- b. On the Salesforce Setup page (in the Classic UI), click the **Contacts** tab.
- c. On the Contacts Home page, click **New**.
- d. On the New Contact page, enter the contact details, such as name, email, phone, address, and so on. Click **Save**.

You've now successfully triggered the recipe, which creates a corresponding user record in Zendesk.

3. Run the recipe to create contacts in Salesforce for Zendesk user records.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow (**Oracle Zendesk Salesforce User2Contact Sync**), then select **Run**.

The Configure and run page is displayed, where you can specify a value for the LastRunTime parameter. This parameter stores the date and time of the most recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Zendesk organization records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the user records from the beginning, set the value as zero.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Zendesk instance and check for the new users created.
 - a. On the Zendesk home page, click **Customers** in the left navigation pane.
 - b. Search for the required organization records on the Customers page.
6. Log in to your Salesforce instance and check for the new contacts created.

- a. On the Salesforce Setup page (in the Classic UI), click the **Contacts** tab.
- b. On the Contacts Home page, select **All Contacts** in the **View** field. Click **Go!** if necessary.
The list of contacts is displayed.
- c. To view a contact record, click its name.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Zendesk Adapter with Oracle Integration 3*

Sync Contracts between SAP Ariba and Oracle ERP Cloud

Use this recipe to synchronize contracts between SAP Ariba and Oracle ERP Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes contracts between SAP Ariba and Oracle ERP Cloud, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it. When triggered, the integration flow reads the contract information from SAP Ariba instance and exports them to Oracle ERP Cloud.

While running the recipe, you can configure the integration flow to read contracts created after a specified time, thereby picking up only the new or required contracts in each run. Basic data such as Contract Name, Agreement Date, Proposed Contract Amount, Effective Date, Expiration Date, and so on are synchronized between SAP Ariba and Oracle ERP Cloud.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- SAP Ariba
- Oracle ERP Cloud
- An account on SAP Ariba with the Administrator role
- An account on Oracle ERP Cloud with the Administrator role

Before You Install the Recipe

Perform the following configuration tasks on your SAP Ariba and Oracle ERP Cloud instances to successfully connect to these external systems using Oracle Integration and synchronize contracts.

Configure SAP Ariba

To access SAP Ariba and export contracts, you must perform certain configuration tasks on your SAP Ariba instance.

To obtain SAP Ariba Host Name and Partition (Site/Realm Name), see Obtain the SAP Ariba Host Name and Partition (Site/Realm Name) in *Using the SAP Ariba Adapter with Oracle Integration 3*.

To configure the Basic Authentication, see Configure Basic Authentication in *Using the SAP Ariba Adapter with Oracle Integration 3*.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the username and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see Assign Required Roles to an Integration User in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user
3. Create a descriptive flexfield to store Ariba Contract Id.
 - a. On the top-right corner of the Oracle ERP Cloud home page, click the profile image, then select **Setup and Maintenance**.
 - b. On the right, click the **Task** icon, then click **Search**.
 - c. In the **Search** field, enter Manage Contract Descriptive Flexfield, and then click the **Search** icon.
 - d. In the Manage Contract Descriptive Flexfield window, click **Search**.
 - e. Select **Contract Header**, and then click the **Edit** icon.
 - f. To create a segment, click the **Create** icon.
 - g. Under Create Segment, enter the name of the flexfield as Ariba Contract Id.
 - h. Under Column Assignment, select the Data Type as **Character**.
 - i. Under Validation, specify the value set. For example: 30 Characters
 - j. Under Display Properties:
 - i. Select the display type as **Text Box**.
 - ii. Specify the display size and display height. For example, 10.
 - k. To enable BI reporting, under Business Intelligence, select the **BI Enabled** checkbox.
4. Deploy the descriptive flexfield. See [Deploy Descriptive Flexfields](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure SAP Ariba

To access SAP Ariba and export contracts, you must perform certain configuration tasks on your SAP Ariba instance.

To obtain SAP Ariba Host Name and Partition (Site/Realm Name), see Obtain the SAP Ariba Host Name and Partition (Site/Realm Name) in *Using the SAP Ariba Adapter with Oracle Integration 3*.

To configure the Basic Authentication, see Configure Basic Authentication in *Using the SAP Ariba Adapter with Oracle Integration 3*.

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORACLE-BRT-ARIBA_ERP_CONTRACT_LOOKUP** lookup table to store values for the Oracle ERP Cloud fields. The lookup table contains data for synchronizing contracts between SAP Ariba and Oracle ERP Cloud.

ERP_Cloud_Field	Description
Default ContractTypeld	An entity used to identify the contract type.

ERP_Cloud_Field	Description
Default OrganizationName	This entry holds the name of the organization.
Default OrganizationId	This entry holds the unique identifier to identify the name of the organization.
Default LegalEntityId	This entry holds the unique identification code of a legal entity.

1. In the Lookups section, click the lookup name.
2. In the **Property_Value** column, enter the name of the ERP_Cloud_Field.
3. In the **Value** column, enter the default values for the corresponding Oracle ERP Cloud fields.
4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe to fetch contracts from SAP Ariba to Oracle ERP Cloud.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the **schedule_param_LastRun** parameter.

- **schedule_param_LastRun**: This parameter stores the date and time of the most recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the contracts created after the date-time stamp stored as the parameter's Current Value are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice not more than past 3 months in the format, yyyy-MM-dd HH:mm.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle ERP Cloud instance and check for the new contracts created.
 - a. Under the **Contract Management** tab, click **Manage Contracts**.
 - b. Search the contract with the **Name** or any other field values.

Related Documentation

- *Using the SAP Ariba Adapter with Oracle Integration 3*
- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*

Sync Customer and Campaign Data Between Oracle Unity and Salesforce Marketing Cloud

Use this recipe to synchronize customer segments and campaign results between Oracle Unity and Salesforce Marketing Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes customer segments and campaign results between Oracle Unity and Salesforce Marketing Cloud. Initially, it exports customer segments from Oracle Unity to Salesforce Marketing Cloud on a scheduled basis. Further, this recipe synchronizes campaign results data between the two systems.

To use the recipe, you must install the recipe and configure the connections and other resources within it.

Initially, you can activate and run the first scheduled integration flow of the recipe (**Oracle Unity FTP Master Customer Export**) manually or specify an execution schedule for it. When triggered, the integration flow fetches a customer segment (with multiple customer records) from Oracle Unity and exports it to a specific location in FTP. Subsequently, run the second scheduled integration flow of the recipe (**Oracle FTP SFMC Data Extension Import**), which reads the customer segment from the FTP location and imports it to a data extension created in Salesforce Marketing Cloud. Finally, run the app-driven integration flow (**Oracle SFMC Automation Setup**) and the last scheduled integration flow of the recipe (**Oracle SFMC Unity Campaign Results Import**) to synchronize the campaign results data between Salesforce Marketing Cloud and Oracle Unity.

Basic data, such as name, email, note, phone, and address, are synchronized between the Oracle Unity and Salesforce Marketing Cloud platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle Unity
- An account on Oracle Unity with the Administrator role
- Salesforce Marketing Cloud
- An account on Salesforce Marketing Cloud with the Administrator role
- A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server

Assumptions

- Salesforce Marketing Cloud API request quota is sufficient to support the number of create and update contact requests from the batch integration.
- Oracle Unity has the customer segment with the required personalization attributes.
- Salesforce Marketing Cloud has the customer data extension already created to which the Oracle Unity customer details must be imported.

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Unity and Salesforce Marketing Cloud instances in order to successfully connect to these external systems using Oracle Integration and synchronize customer and campaign data between them.

Configure Oracle Unity

Complete the following tasks in your Oracle Unity instance to successfully connect to it from Oracle Integration and export customer segments.

1. Obtain the client ID and client secret. See *Prerequisites for Creating a Connection in Using the Oracle Unity Adapter with Oracle Integration 3*.
2. Create master customer segments. See [Creating Segments](#) in Oracle Unity documentation.
3. Add the following personalization attributes to your master customer segments. See [Managing Personalization Attributes](#) in Oracle Unity documentation.
 - ID, First Name, Last Name, Email, Phone, Country, Age, Gender, Ok To Email, and Ok To Text.

Access Your FTP Server and Create a Directory

Obtain an SFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server to import and archive the files. Note the path of this directory.

Configure Salesforce Marketing Cloud

To access Salesforce Marketing Cloud from Oracle Integration and import customer segments, you must perform certain configurations on your Salesforce Marketing Cloud instance.

- Obtain the client ID and client secret for a new package.
- Obtain the client ID and client secret for an existing package.
- Create an import data extension and obtain the external key.
- Obtain the SOAP WSDL endpoint, username, and password.

Log in to your Salesforce Marketing Cloud instance as an **Administrator** and execute the following tasks.

1. Obtain the client ID and client secret for a new package.

- a. Click **Setup**, select **Apps**, and then select **Installed Packages**.
 - b. Click **New** to create a new package.
 - c. In the New Package Details window, enter the name and description for the package and click **Add Component**.
 - d. Select **API Integration** as the component type and click **Next**.
 - e. Select **Server-to-Server** as the integration type and click **Next**.
 - f. In Server-to-Server properties, select **Read and Write Properties** for the Data Extensions property.
 - g. Click **Save**.
The Components section displays the **Client ID** and **Client Secret** for the package. Note these values.
2. Obtain the client ID and client secret for an existing package.
 - a. Click **Setup**, select **Apps**, and then select **Installed Packages**. The Installed Packages page lists all the packages installed in the Salesforce Marketing Cloud account.
 - b. Select a package and navigate to the Components section.
 - i. To add a new component, click **Add Component**.
 - ii. To edit an existing component, click **Edit**.
The Components section displays the **Client ID** and **Client Secret** for the existing package. Note these values.
 - c. Click **Save**.
3. Create an import data extension and obtain the external key.
 - a. In **Audience Builder**, choose **Contact Builder**.
 - b. In the Contact Builder window, navigate to the **Data Extensions** tab, and click **Create**.
 - c. Select the appropriate data extension method in the **Creation Method** drop-down menu.
 - i. **Create from New** – To create a data extension and manually insert all attributes.
 - ii. **Create from Existing** – To create a data extension and include all attributes assigned through the existing data extension or add attributes later in the creation process.
 - iii. **Create from Template** – To create a data extension and include all attributes assigned in the template or add attributes later in the creation process.
 - d. Enter a name for the data extension.
 - e. Note the value in the **External Key** field. Each data extension uses a unique external key.
 - f. Enter a brief description of the data extension.
 - g. Click **Change Location** and choose the location for storing the data extension. Choose an existing folder or create a new folder.
 - h. Select the **Is Sendable?** check box to make this data extension available as the source.
 - i. Select the **Is Testable?** check box to make this data extension available for testing and click **Next**.

- j. Set the data retention policy as **OFF**, then click **Next**.
- k. Create attribute fields to match the fields with Oracle Unity. For example, Unity_MasterCustomer_ID, FirstName, LastName, Email, Phone, Country, Age, Gender, OkToEmail, and OkToText.

Note

For the Unity_MasterCustomer_ID attribute, select the **Primary Key** check box.

For each attribute:

- Enter a name and choose a data type.
- If the attribute must contain a value for every contact, choose **Required**.
- Enter the maximum number of characters the attribute can have in the **Length** field.
- If the attribute must have a default value, enter the value in the **Default Value** text field.
- For sendable data extensions, map the **Send relationship** from Unity_MasterCustomer_ID data extension field to the **Subscriber Key**.

Note

Attributes assigned as a primary key do not contain default values.

- l. Click **Complete**.
4. Obtain the SOAP WSDL endpoint, username, and password. To do so, you must create a user and assign role and permissions.
 - a. Create a user.
 - i. Click **Setup**, select **Users**, and then **Users**.
 - ii. On the Setup Users page, click **Create**.
 - iii. On the New User page:
 - Enter the name of the user.
 - Enter the email ID of the user in **Reply Email Address** field and verify it.
 - Enter the email address to which run-time exception emails are sent in the **Notification Email Address** field.
 - Enter the username and note the same.
 - Select the appropriate **Time Zone** and **Culture Code**.
 - Enable the **API User** option.
 - Note the **SOAP WSDL URL**.
 - Set a password and note the same.
 - b. Assign the required role and permissions to the user.
 - i. On the Setup Users page, select the user you created previously.

- ii. Click **Manage Roles**.
- iii. Select whether you assign the role directly to the user or in the context of a specific business unit in the **Business Unit** drop-down menu.
- iv. Click **Edit Roles**. Select the check box next to the appropriate roles to assign.
- v. Click **Save**.
- vi. Click **Edit Permissions** to update specific permissions for the user and select the check boxes next to permissions to assign.
- vii. Click **Save**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Unity Connection

1. In the Connections section, click the connection name.
2. In the **Oracle Unity URL** field, enter the URL for connecting to Oracle Unity. For example, <https://XXXXXX.cxunity.ocs.oraclecloud.com>.

The URL is provided when you subscribe to Oracle Unity.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Resource Owner Password Credentials .
Access Token URI	Enter the authorization server that generates the access token: https://IDCS_URL/oauth2/v1/token
Client ID	Enter the unique random string that matches the API console project. See Configure Oracle Unity .
Client Secret	Enter the unique random string that matches the API console project.
Username	Enter the Oracle Identity Cloud Service username of the OAuth token request.
Password	Enter the Oracle Identity Cloud Service password of the OAuth token request.
Scope	Enter the list of authorization permissions for the target application. For example, read write.

4. Click **OK**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Salesforce Marketing Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Paste the URL without trailing (/) copied from Salesforce Marketing Cloud. See Configure Salesforce Marketing Cloud .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Custom Two-Legged Flow .

Field	Information to Enter
Access Token Request	Enter the client id, client secret, account id and auth url copied from Salesforce Marketing Cloud in the below format. Append v2/token in the auth url. To generate client id and secret, see Configure Salesforce Marketing Cloud . -X POST -H 'Content-Type:application/json' -d '{ "grant_type": "client_credentials", "client_id": "<client id>", "client_secret": "<client secret>", "account_id": "<account id>" }' 'https://mc4nwk3rs1n1tm8lm4qgd1g448hm.auth.marketingcloudapis.com/v2/token'

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the SOAP Salesforce Marketing Cloud Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the SOAP WSDL URL in the **WSDL URL** field. See [Configure Salesforce Marketing Cloud](#).
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username. See Configure Salesforce Marketing Cloud .
Password	Enter the password.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the REST OIC Invoke Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Paste the URL of the Oracle Integration instance in the below format.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of the Oracle Integration instance.
Password	Enter the password of the Oracle Integration instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST OIC Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, select the Security Policy as **OAuth 2.0 Or Basic Authentication**.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

The recipe contains the following four lookup tables. Edit the lookup tables to enter appropriate values for the lookup keys.

- **ORACLE-BRT-UNITY_SFMC_PROPERTIES**: Used to configure the FTP file details in which the customer segments are stored.

Key	Description	Example
FileDirectory	Stores the location of the FTP file directory that has the customer segments.	/home/users/org-name / customers/inbound
RetryCount	Stores the number of retries when a transient error occurs.	3
ArchiveEnabled	Stores the flag to identify if the file must be archived after being processed successfully.	Y
ArchiveDirectory	Stores the location of the FTP directory that stores the archived files with customer segments.	/home/users/org-name / customers/archive
CustomerDataExtensionKey	Stores the ID of customer data extension in Salesforce Marketing Cloud. It is a unique value that identifies the data extension and is used to identify the data extension with an API call.	5C2EF345-4A12-405C-BDCE-BE358139DC44

Key	Description	Example
SFMCPageSize	Stores the ideal pagination value for the Oracle FTP SFMC DataExtension Import integration.	250
EmailFrom_Success	Stores the email address from which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	no-reply@company.com
EmailTo_Success	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	xyz@company.com
EmailSubject_Success	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	Unity to SFMC Customer Import Successful
EmailFrom_Failure	Stores the email address from which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	no-reply@company.com
EmailTo_Failure	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	xyz@company.com
EmailSubject_Failure	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	Unity to SFMC Customer Import Failure
MaxFileCount	Stores the maximum number of files that can be processed in one instance. Ensure that the processing time doesn't exceed six hours limit for a scheduled instance. The maximum possible value is 100.	100
MaxInstanceCount	Stores the maximum number of instances that a run can span across in case the processing is not completed in one instance.	10
 Note Ensure that processing is complete before next scheduled run.		
ClearBeforeImport	Stores the flag to identify if recipe clears the old data before importing the new data in the customer import data extension.	Y
CustomerSegmentName	Stores the Customer Segment Name.	Unity_SFMC_MasterCustomer

- **ORACLE-BRT-SFMC_UNITY_PROPERTIES:** Used to configure the data extension that stores the campaign results in Salesforce Marketing Cloud.

Key	Description	Example
UnityTenantId	Stores the Unity Tenant Id.	100046
ResultsDataExtensionKey	Stores the external key of the results data extension in Salesforce Marketing Cloud. This value is passed in the Automation Setup integration.	ResultsDataExtensionName
ResultsDataExtensionName	Stores the name of the results data extension in Salesforce Marketing Cloud. This value is passed in the Automation Setup integration.	Unity_CampaignResults_DE
MarketingMedium	Stores the medium of the marketing campaign.	Email
MarketingSource	Stores the source application of the marketing campaign.	Salesforce Marketing Cloud
EmailFrom_Success	Stores the email address from which a message is sent whenever the campaign results import from Salesforce Marketing Cloud to Oracle Unity is successful.	no-reply@company.com
EmailTo_Success	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	xyz@company.com
EmailSubject_Success	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	Unity to SFMC Customer Import Successful
EmailFrom_Failure	Stores the email address from which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	no-reply@company.com
EmailTo_Failure	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	xyz@company.com
EmailSubject_Failure	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	Unity to SFMC Customer Import Failure
UnityPageSize	Stores the maximum number of records that can be passed to Oracle Unity in an invoke request.	500
UnityRetryCount	Stores the maximum number of retries in case Oracle Unity invoke request fails.	3

Key	Description	Example
MaxIterationCount	Stores the maximum number of Salesforce Marketing Cloud iterations with an iteration size of 2500 records. Ensure that the processing time doesn't exceed six hours limit for a scheduled instance.	100
MaxInstanceCount	Stores the maximum number of instances that a run can span across in case the processing is incomplete in one instance.	20

 Note

Ensure that the processing gets completed before next scheduled run.

- **ORACLE-BRT-SFMC_UNITY_EVENT_TYPE**: Represents the mapping between event types in the instances. For instance, SentEvent is mapped with Sent.

Event Type in Salesforce Marketing Cloud	Event Type in Oracle Unity
SentEvent	Sent
OpenEvent	Opened
ClickEvent	Clicked
BounceEvent	Bounced

- **ORACLE-BRT-SFMC_UNITY_MONTH_OF_YEAR**: Used to map the calendar months between the instances.
 1. In the Lookups section, click the lookup name.
 2. Edit the lookup table as necessary.
 3. Click **Save**. If prompted, click **Save** again.
 4. To return to the project workspace, click **Go back** .

Add or Edit the Attributes in Salesforce Marketing Cloud (Optional)

To add or edit attributes in Salesforce Marketing Cloud (other than what is pre-configured in this recipe), you must update the data extension you created previously and update the schema in REST endpoint configuration of the **Oracle FTP SFMC DataExtension Import** integration flow.

Update the Data Extension in Salesforce Marketing Cloud

1. Log in to Salesforce Marketing Cloud.
2. In **Audience Builder**, select **Contact Builder**.
3. In the Contact Builder window, navigate to the **Data Extensions** tab.
4. Select the data extension you created previously from the list, then click **Open**.
5. To add a new attribute, click **Create Attribute**.

- To edit an attribute, click the **Edit** icon next to the attribute.

Note

You cannot select the data type while editing the attribute.

- Click **Save**.

Update the Schema in REST Endpoint Configuration

- In the Integrations section, click the **Oracle FTP SFMC DataExtension Import** integration flow.
- On the integration canvas, expand **FileScope**, then expand **PageScope**.
- In PageScope, select the **UpsertInSFMCDataExtension** element, and then click **Actions** and select **Edit**.
- On the Request page, update the JSON schema with the new fields.
- On the Summary page, click **Done**.
- In the resulting window, click **Update**.
The updated attribute appears as a separate field on the mapping canvas.

Update the REST Endpoint's Mapper

Map the new attribute to the relevant source attribute on the mapping canvas.

- On the integration canvas, in **PageScope**, select the **Map to UpsertInSFMCDataExtension** element, then click **Actions** and select **Edit** to open the mapper.
- Map the new attribute to the relevant source attribute on the mapping canvas.

Add or Delete Attributes in Oracle Unity (Optional)

To add or delete attributes in Oracle Unity (other than what is pre-configured in this recipe), you must perform the following tasks:

- Add or delete the personalization attributes in Oracle Unity.
- In the **Oracle FTP SFMC DataExtension Import** integration flow of the recipe:
 - Update the FTP adapter with the new file schema.
 - Update the mapper element corresponding to the FTP adapter.
 - Update the Stage File element with the new file schema.
 - Update the mapper element corresponding to the Stage File element.
 - Update the Stage File element in the fault handler with the new file schema.
 - Update the mapper element corresponding to the Stage File element in the fault handler.

Add or Delete the Personalization Attributes in Oracle Unity

To add or delete the personalization attributes (other than what is pre-configured in this recipe) from the base object to the segment, see [Managing Personalization Attributes](#) in Oracle Unity documentation.

Update the FTP Adapter with the New Schema

Note

You get the updated file schema when you run the **Oracle Unity FTP Master Customer Export** integration flow.

1. In the Integrations section, click the **Oracle FTP SFMC DataExtension Import** integration flow.
2. On the integration canvas, expand the **FileScope** element, select the **ReadFileFromFTP** element, and then click **Actions** . . . and select **Edit**.
3. Upload the updated schema.
4. On the Summary page, click **Done**.
5. In the resulting window, click **Update**.
The updated attribute appears as a separate field on the mapping canvas.

Update the FTP Adapter's Mapper

1. On the integration canvas, in **FileScope**, select the **Map to ReadFileFromFTP** element, and then click **Actions** . . . and select **Edit** to open the mapper.
2. Map the new attribute to the relevant target attribute on the mapping canvas.

Update the Stage File Element with the New Schema

1. On the integration canvas, in **FileScope**, select the **InitializeStageErrorFile** element, and then click **Actions** . . . and select **Edit**.
2. Upload the updated schema.
3. On the Summary page, click **Done**.
4. In the resulting window, click **Update**.
The updated attribute appears as a separate field on the mapping canvas.

To know more about configuring stage file activity, see *Configure a Stage File Action in Using Integrations in Oracle Integration 3*.

Update the Stage File Element's Mapper

1. On the integration canvas, in **FileScope**, select the **Map to InitializeStageErrorFile** element, and then click **Actions** . . . and select **Edit** to open the mapper.
2. Map the new attribute to the relevant target attribute on the mapping canvas.

Update the Fault Handler's Stage File Element with the New Schema

1. On the integration canvas, expand **FileScope**, and then expand **PageScope**.
2. In **PageScope**, click the **Fault Handler** element, and select **Default Handler**.
3. Select the **AppendToStageErrorFile** element, and then click **Actions** . . . and select **Edit**.
4. On the Format Definition page, upload the updated schema.
5. On the Summary page, click **Done**.
6. In the resulting window, click **Update**.
The updated attribute appears as a separate field on the mapping canvas.

To know more about configuring stage file activity, see *Configure a Stage File Action in Using Integrations in Oracle Integration 3*.

Update the Stage File Element's Mapper

1. On the integration canvas, within **DefaultHandler**, select the **Map to AppendToStageErrorFile** element, and then click **Actions** . . . and select **Edit** to open the mapper.
2. Map the new attribute to the relevant target attribute on the mapping canvas.

Add a New Email Campaign Event (Optional)

In this recipe, the following email campaign events are pre-configured:

- Sent Event
- Bounce Event
- Click Event
- Open Event

To add a new email campaign event other than what is pre-configured in the recipe, you must:

1. Add a new event in the lookup table **ORACLE-BRT-SFMC_UNITY_EVENT_TYPE**. See [Configure the Lookup Tables](#).
2. In the Integration section, click the **Oracle SFMC Automation Setup** integration flow and perform the following tasks on the integration canvas.
 - a. Assign global variables for the new event.
 - i. On the integration canvas, select the **AssignGlobalVariables** element, and then click **Actions** . . . and select **Edit**.
 - ii. Add new variables for the event.
 - b. Add a SOAP adapter endpoint and a mapper for the new event.
 - i. In the **Main Scope**, expand **DataExtensions Scope**.
 - ii. Add a SOAP adapter endpoint along with a mapper element.
 - iii. Update the mapper with the details specific to the new event.
 - c. Add a status check for the data extension created.
 - i. In the **DataExtensions Scope**, select the switch case (**IF DataExtensionsCreated**) element, and then click **Actions** . . . and select **Edit**.
 - ii. Add a **StatusCode** for the new data extension created.
 - d. Add query definitions for the event.
 - i. In the **Main Scope**, expand **QueryDefinitions Scope**.
 - ii. Add a SOAP adapter endpoint and a mapper for a query definition to query events data from the Salesforce Marketing Cloud data view and import the same to the data extension created for the event.
 - iii. Update the query definition mapper with the details and query specific to the new event.
 - iv. Add another SOAP adapter endpoint and a mapper for a query definition to copy data from the event-specific data extension to the results data extension.
 - v. Update the query definition mapper with the details and query specific to the new event.
 - vi. Add switch case conditions and assign variables for the data extensions created.

- i. In the **QueryDefinitions Scope**, select the switch case (**IF QueryDefinitionsCreated**) element, and then click **Actions** ... and select **Edit**.
 - ii. Add a **StatusCheck** for the new data extension created.
 - iii. To assign variables, select the **AssignQueryDefinition** element, and then click **Actions** ... and select **Edit**.
 - iv. Add a new variable in the similar format you added for other events.
- e. In the **Automation Scope**, update the mapper with the query definition created.
 - i. In the **Main Scope**, expand **AutomationScope**.
 - ii. Select the **Map to createAutomation** element, and then click **Actions** ... and select **Edit**.
 - iii. On the mapping canvas, add a new **AutomationTask** and map the new query definition created with the relevant target attribute.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Export a customer segment from Oracle Unity to an FTP server.
 - i. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle Unity FTP MasterCustomer Export** integration flow, then select **Run**.
 - ii. On the Configure and run page, click **Run**.
The integration gets triggered, and the customer segment records are exported to the FTP server.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

- b. Import a customer segment from the FTP server to Salesforce Marketing Cloud.
 - i. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle FTP SFMC DataExtension Import** integration flow, then select **Run**.
The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.
 - **continuelInstance**: This parameter stores the flag to identify a new instance. Y indicates continued process from previous instance. N indicates a new instance and is the default value.
 - **InstanceNo**: This parameter stores the count of the running instance. The default value is 1.
 - **triggerTimeStamp**: This parameter stores the created time stamp of the trigger file that started the instance.
 - ii. On the Configure and run page, click **Run**.

The integration gets triggered, the customer segment data is imported to Salesforce Marketing Cloud data extension.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

- c. Run the **Oracle SFMC Automation Setup** integration flow to create the initial customization required in Salesforce Marketing Cloud. This is a one-time setup that configures the automation that queries and stores campaign results in a data extension.

- i. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, and select **Run**, then click **Test**.

The page to test the integration with a sample file is displayed.

- ii. In the Request section of the test page, on the **Body** tab, provide the following parameters:

Name	Description	Example
customerDataExtensionName	Stores the name of the data extension to which customer segments are imported from Oracle Unity to Salesforce Marketing Cloud.	Unity_MasterCustomer_DE
automationKey	Stores the external key of the automation created in Salesforce Marketing Cloud.	Unity_CampaignResults_Automation
automationName	Stores the name of the automation created in Salesforce Marketing Cloud.	Unity_CampaignResults_Automation
resultsDataExtensionKey	Stores the external key of the data extension with the campaign result events in Salesforce Marketing Cloud.	Unity_CampaignResults_DE
resultsDataExtensionName	Stores the name of the data extension with the campaign result events in Salesforce Marketing Cloud.	Unity_CampaignResults_D
automationScheduleStartDateTime	Stores the DateTimeStamp when the automation schedule starts in CST Time Zone.	YYYY-MM-DDT00:00:00
automationScheduleOccurrences	Stores the number of occurrences of the automation schedule runs.	1000

Note

The automation runs on a daily schedule.

automationScheduleInterval	Stores the number of days interval between two occurrences of the scheduled run.	1

- iii. Click **Test**.

- d. Synchronize campaign results data from Salesforce Marketing Cloud to Oracle Unity.

- i. In the Integrations section of the project workspace, click **Actions** on the **Oracle SFMC Unity CampaignResults Import** integration flow, then select **Run**. The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.
 - **continueInstance**: This parameter stores the flag to identify a new instance. Y indicates continued process from previous instance. N indicates a new instance and is the default value.
 - **InstanceNo**: This parameter stores the count of the running instance. The default value is 1.
 - **continueRequestId**: This parameter stores the Request ID of the Salesforce Marketing Cloud request which needs to be continued in upcoming instances.
- ii. On the Configure and run page, click **Run**. The integration gets triggered, and the email campaign results are imported to Oracle Unity.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to Oracle Unity and check for the new campaign results imported.

Related Documentation

- [Using the Oracle Unity Adapter with Oracle Integration 3](#)
- [Using the FTP Adapter with Oracle Integration 3](#)
- [Using the Salesforce Adapter with Oracle Integration 3](#)

Sync Data Between Azure Storage and PostgreSQL Upon Receiving Notification on Azure Event Grid

Use this recipe to synchronize data between Azure Storage and PostgreSQL upon receiving notification on Azure Event Grid.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how a file that is uploaded to Azure Storage gets reflected in PostgreSQL database when a notification is received on Azure Event Grid.

In this recipe, the Event grid schema is used from Azure Event Grid. GET Blob (file) is used to read the uploaded file by Azure Storage (Invoke) and Put Blob (file) is used to upload the file

into Azure Storage (Invoke). The Data Import Operation is reflected in the PostgreSQL database (Invoke).

To use the recipe, you must install the recipe and configure the connection and other resources within the recipe. The recipe uses three adapters in two integration flows. The first integration flow (**Oracle FTP Azure Storage Create Data**) is a scheduled orchestration style integration which reads a file from FTP to upload the data in Azure Storage. The second integration flow (**Oracle AEG Azure Storage PostgreSQL Data Sync**) is an app-driven integration that syncs data from Azure Storage to PostgreSQL on receiving a notification on Azure Event Grid.

System and Access Requirements

- Oracle Integration 3
- Azure Event Grid for creating subscriptions to receive notifications, with the following roles:
 - Azure Event Grid Reader
 - Azure Event Grid Contributor
 - Azure Event Grid Sender
- Azure Storage for uploading the files, with the following roles:
 - Data Reader
 - Data Contributor
- A PostgreSQL database
- An account on PostgreSQL database with the Administrator role, for importing the data.
- A secure FTP (sFTP) server or File Server.
- An FTP client to read the data/file from the server.

Before You Install the Recipe

You must perform the following configuration tasks on your FTP, Azure Event Grid, Azure Storage and PostgreSQL instances to successfully connect to these external systems using Oracle Integration and synchronize data between Azure Storage and PostgreSQL.

Access Your FTP Server

Obtain an FTP server and ensure that you complete the **FTP Server Prerequisites** listed in Prerequisites for Creating a Connection in *Using the FTP Adapter with Oracle Integration 3*.

Also, ensure that you have the read and write permissions on the FTP server directory to and from which you want to transfer files.

Configure Azure Event Grid

To access Azure Event Grid from Oracle Integration and successfully receive notifications on created subscriptions you must complete certain prerequisites.

See Prerequisites for Creating a Connection in *Using the Azure Event Grid Adapter with Oracle Integration 3*.

Configure Azure Storage (Get Blob)

To successfully get a file to Azure Storage using Oracle Integration, Azure Storage needs to get the blob for respective container.

See Prerequisites for Creating a Connection in *Using the Azure Storage Adapter with Oracle Integration 3*.

Configure PostgreSQL

To access the PostgreSQL database from Oracle Integration and synchronize data, you must perform certain configurations on your PostgreSQL instance.

For general configuration tasks, see Prerequisites for Creating a Connection in *Using the PostgreSQL Adapter with Oracle Integration 3*.

In addition, you must create a database table. To create a database table:

1. Log in to your PostgreSQL Workbench instance as an **Administrator**.
 2. Create the database table in either of the two ways given below.
 - a. Right-click on **Public Schema**.
 - b. Click **Query Tool**.
 - c. Paste the query on the Query tool and then execute.
For a sample query, refer to the following: [PostgreSQLTableCreateQuery.txt](#)
- or,
- a. Under public schema, right-click **Tables**, select **Create**, and then **Tables**.
 - b. In the Create-Table dialog, specify the name of the table as aeg.
 - c. Right-click on the empty table, and under **Create** select **Column**.
 - d. Click **Save**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Azure Event Grid Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Subscription ID	Enter a valid Subscription ID. See Configure Azure Event Grid .
Resource group	Enter a valid Resource Group.
API version	Enter a valid API version.

3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter a valid Client ID. See Configure Azure Event Grid .
Client Secret	Enter a valid Client Secret.
Tenant ID	Enter a valid tenant ID.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Azure Storage Invoke Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Storage Account name	Enter a valid storage name. See Configure Azure Storage (Get Blob) .
Tenant Id	Enter a valid tenant ID.

3. In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter a valid Client ID. See Configure Azure Storage (Get Blob) .
Client Secret	Enter a valid Client Secret.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the PostgreSQL Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Server Name	Enter a valid server name. See Prerequisites for Creating a Connection in <i>Using the PostgreSQL Adapter with Oracle Integration 3</i> .
Port	Enter a valid port number.
DBNAME	Enter a valid database name.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	By default, a security policy is selected.
Username	Enter the username to connect to the database. See <i>Prerequisites for Creating a Connection in Using the PostgreSQL Adapter with Oracle Integration 3</i> .
Password	Enter the password to connect to the database.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host	Enter a valid host address of your FTP server.
Port	Enter a valid port number.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	By default, a security policy is selected.
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update integration property values for the **Oracle FTP Azure Storage Create Data** and **Oracle AEG Azure Storage PostgreSQL Data Sync** integration flows.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.

- b. In the Update property values panel, update the integration properties with appropriate values.
 - **EmailTo**: This integration property holds the email ID to which error notifications are to be sent.
 - **retryCount**: This integration property holds the number of retries when an invoke fails. The default value is 3.
 - c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.
3. Run the **Oracle FTP Azure Storage Create Data** integration flow.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The integration now reads a file from FTP and creates the data in Azure Storage.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

4. Run the **Oracle AEG Azure Storage PostgreSQL Data Sync** integration flow.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The integration now synchronizes data from Azure Storage to PostgreSQL on receiving a notification on Azure Event Grid.

Note

To run the recipe, you must create a subscription in Azure Event Grid and a database table in PostgreSQL.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Log in to the Azure Event Grid instance and check for the notification received.
 - a. Click the wired message.
 - b. Click the eye icon to open the message.
 - c. Observe the Blob-created message.

Thus, you have successfully received a notification on uploading a file.

7. Log in to the PostgreSQL instance, and check for the data created.
 - a. On the DB visualizer, create a successful connection.
 - b. Enter the query as `select * from tablename;`
 - c. Observe if the data has been imported successfully.
 - d. Cross-verify the file data and imported data into the database.

Thus, you have successfully imported data from Azure Storage into PostgreSQL database.

Related Documentation

- *Using the Azure Event Grid Adapter with Oracle Integration 3*
- *Using the Azure Storage Adapter with Oracle Integration 3*
- *Using the PostgreSQL Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Sync Data Between Oracle CPQ Transactions and Salesforce Quotes

Use this recipe to synchronize data between Oracle Configure, Price, Quote (Oracle CPQ) transactions and Salesforce quotes.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

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 and configure the connections and other resources within the recipe. 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

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

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.

1. 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 3*.

2. 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*.

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.

- b. In the Admin navigation pane, expand **Users** and click **Internal Users**.

The User Administration List page appears.

- c. Add the user from the User Administration List page.

For 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**.

- e. In the resulting page enter the following details, and click **Test** to verify if your connection works.

Field	Information to enter
Name	Enter a suitable name.
Variable 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.
Discovery URL	Enter the Oracle Integration instance URL.
Username	Enter the username to access your Oracle Integration instance.
Password	Enter the password to access your Oracle Integration instance.

- 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**.
7. 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**.
- b. 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.

① Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and 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**.
 - b. In the resulting page, click **New User**.
 - c. Create a username and password.
 - i. 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**.
 - 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.
 - 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 Lightning 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 Procure Book page's URL. For example, if the URL is `https://<instance name>.my.salesforce.com/01s0075956A1eUQ`, then the Salesforce ID is `01s0075956A1eUQ`.

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.
 - 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, <code>TransactionId_CPQ</code> .
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, <code>TransactionId_CPQ</code> .
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.

Field	Information to Enter
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.

i. 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.

Field	Information to Enter
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 <i>LastModifiedBy.CommunityNickname</i> and select it.
Operator	Select Does not equal .
Type	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 <i>LineItemCount</i> and select it.
Operator	Select Greater than .
Type	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.
Type	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.
Type	Select Field Reference .
Value	Click this field and on the resultant Select a Field dialog, select TransactionId_CPQ from the drop-down, 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle CPQ REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST Catalog URL .
Connection URL	Enter the instance URL of Oracle CPQ. For example, <code>https://OracleInstance.com/rest/v8/metadata-catalog</code> . See Configure Oracle CPQ .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration 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 .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle CPQ SOAP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select SOAP WSDL URL .
Connection URL	Enter the URL of the Oracle CPQ instance. For example, <code>https://OracleInstance.com/rest/v8/metadata-catalog</code> . See Configure Oracle CPQ .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration 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 .

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Salesforce Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. See step 3 in Configure Salesforce .

- 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.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Configure the lookup table to replace the default Salesforce ID value with the specific Salesforce ID for the Standard Book Price of your Salesforce instance.

1. In the Lookups section, click the lookup name.

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.

2. 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](#).

3. Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Configure the Integration Flow

Configure the integration so that Oracle Integration subscribes to Salesforce platform events that are published, and processes the appropriate platform event that you have created for this recipe.

1. In the Integrations section, click the integration flow name.

2. Click the Salesforce trigger (GetQuote_PE), and then click **Edit** .

The Configure Salesforce.com Endpoint wizard is displayed.

3. Click **Next**.

4. On the Salesforce Trigger Action page, select **Streaming API**, and click **Next**.

5. On the Streaming API Configuration page, select **Subscribe to platform events published on Salesforce.com**.

6. In the **Select a platform event** text box, select the platform event that you created earlier.

See step 7 in [Configure Salesforce](#).

7. Click **Next**.

The Salesforce.com Endpoint Configuration Summary dialog appears.

8. Click **Done**.

9. Click **Save**. If prompted, click **Save** again.

10. Click **Close**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

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. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Test the Oracle CPQ to Salesforce integration flow.
 - a. Create a transaction in Oracle CPQ.
 - i. Log in to the Oracle CPQ 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 Item 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 running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
 - c. Check if a corresponding quote is created in Salesforce.
 - i. Log in to your Salesforce instance.

Note

If you're shown the Lightning 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.

Note

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 integration flow.
 - a. Create a quote with line-items in Salesforce.
 - i. Log in to your Salesforce instance.

Note

If you're shown the Lightning 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 running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
- c. Check if the corresponding transaction is created in Oracle CPQ.
 - 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 3*
- *Using the Salesforce Adapter with Oracle Integration 3*

Sync Email Accounts Between Okta and Oracle HCM Cloud

Use this recipe to synchronize email accounts between Okta and Oracle HCM Cloud, where Okta is the master.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to use Oracle Integration to synchronize email accounts between Okta and Oracle HCM Cloud, where Okta is the master.

The integration assumes that Okta is acting as a broker for an email service. Initial workers are created within Oracle HCM Cloud and loaded into Okta. Once created in Okta, an email service creates a new email, and updates the value within Okta, through an Okta connector. Oracle Integration retrieves the updated email address and then makes an update call into Oracle HCM Cloud.

To use the recipe, you must install the recipe package and configure the connections 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, this integration flow synchronizes email accounts between Okta and Oracle HCM Cloud.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle HCM Cloud, Release 13 or later
- An account on Oracle HCM Cloud with the Administrator role
- Okta instance
- An account on Okta with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle HCM Cloud and Okta instances in order to successfully connect to these external systems using Oracle Integration and synchronize email accounts.

Configure Oracle HCM Cloud

To access the Oracle HCM Cloud instance from Oracle Integration and synchronize email accounts, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud, and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Configure Okta API Key

To access the Okta instance from Oracle Integration, you'll require the API key associated with your Okta account.

To obtain your Okta account's API key:

1. Log in to your Okta instance as an administrator.
2. Click **Admin** to open the Admin console.
3. From the menu, select **Security** and choose **API**.
4. Navigate to the **Tokens** tab.
5. Note the API Token.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Connection Properties section, enter the Oracle HCM Cloud host name.
For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Configure Oracle HCM Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the REST Okta Connection

1. In the Connections section, click the connection name.
2. In the Connection Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave REST API Base URL selected.
Connection URL	Specify your Okta URL. For example: <code>example.okta.com</code> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave API Key Based Authentication selected.
API Key	Enter the API key associated with your Okta account. See Configure Okta API Key .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle HCM Cloud instance and check for the updated email addresses.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*

Sync Employee Identities Between Oracle HCM Cloud and an LDAP Instance

Use this recipe to synchronize and manage the life cycle for employee identities between Oracle HCM Cloud and an LDAP instance.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe explores how you can use Oracle Integration to create LDAP Data Interchange Format (LDIF) files for employees within your Oracle HCM Cloud instance.

The integration covers the following scenarios:

- New hires
- Updates to existing employees records (employee details and assignments)
- Employee terminations

The integration, which is complex, contains each scenario within a separate scope. You can customize and extend the integration for use within your company.

System and Access Requirements

- Oracle Integration
- Oracle HCM Cloud
- LDAP instance

Before You Install the Recipe

To access the Oracle HCM Cloud instance from Oracle Integration and extract employee identities, you'll need to create a user account for Oracle Integration on Oracle HCM Cloud,

and make a few other configurations. See *Prerequisites for Creating a Connection in Using the Oracle HCM Cloud Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle HCM Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle HCM Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on Oracle HCM Cloud. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on Oracle HCM Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the FTP Read Write Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and map the attributes between the Oracle HCM Cloud instance and your LDAP instance.

1. In the Lookups section, click the lookup name.
2. Map the attributes in the Oracle HCM Cloud instance to the corresponding attributes in your LDAP instance.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your FTP server and check for the employee identities fetched from your Oracle HCM Cloud instance.

Related Documentation

- *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*

Sync ESG Data from EcoVadis to Suppliers in Oracle Procurement Cloud

Use this recipe to synchronize environment, social and government (ESG) data from EcoVadis to suppliers in Oracle Procurement Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an event-driven recipe that synchronizes ESG data from EcoVadis to Oracle Procurement Cloud suppliers that are created during supplier registration.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When a new supplier is created from registration in Oracle Procurement Cloud, a business event is raised. The first integration flow, **Supplier EcoVadis Request ESG Score** is triggered by the event and a request is sent to EcoVadis. This request contains the key information about the supplier including supplier number which is used as the integration id to uniquely identify the supplier in the second integration flow. EcoVadis will use the information in the request to locate the company on its platform or launch the assessment process for ESG scores if needed. The second integration flow, **Supplier EcoVadis Sync ESG Score** is run regularly (preferably daily) to retrieve the latest ESG data from EcoVadis and update the supplier profiles with matching integration ids.

System and Access Requirements

- Oracle Integration, Version 24.06 or higher
- Oracle Procurement Cloud
- An account on Oracle Procurement Cloud with Supplier Manager and Application Administrator roles
- EcoVadis

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Procurement Cloud and EcoVadis instances in order to successfully connect to these external systems using Oracle Integration, and sync ESG data from EcoVadis to Oracle Procurement Cloud suppliers.

Configure Oracle Procurement Cloud

To access the Oracle Procurement Cloud instance from Oracle Integration you'll require a separate user account on Oracle Procurement Cloud.

Log in to your Oracle Procurement Cloud instance as a **Application Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle Procurement Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information on these roles, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user
3. Create a custom role by copying **Supplier Manager** (ORA_POZ_SUPPLIER_MANAGER_ABSTRACT) role and assign the privilege **Update Supplier Pending Spend Authorization Using REST Service** (POZ_UPDATE_SUPPLIER_PENDING_SPENDAUTH_VIA_REST_PRIV) to the custom role. See [Role Copying or Editing](#). This will allow updating supplier profile with EcoVadis ESG data.
4. Assign the custom role to the user account created above.
5. Configure supplier descriptive flexfields:
 - a. Navigate to the Setup and Maintenance work area.
 - b. Search for the setup task Manage Supplier Descriptive Flexfields and click **POZ_SUPPLIERS**.
 - c. Click **Edit**.
 - d. Refer to the following table to configure global segments where the ESG data from EcoVadis will be stored on the supplier profile.

Note

Ensure that you specify the values given in the table below for **API Name**, **Data Type** and **Value Set**.

For other fields, you can enter values according to your specific requirements.

Name	Code	API Name	Data Type	Table Column	Value Set	Prompt	Display Type	Readonly
EcoVadis Current Stage	EV_Current_Stage	EVCurentStage	Character	POZ_SUPPLIER_S_ATTRIBUT16	60 Character	EcoVadis Current Stage	Text Box	Y
EcoVadis Progress Status	EV_Progress_Status	EVProgressStatus	Character	POZ_SUPPLIER_S_ATTRIBUT17	60 Character	EcoVadis Progress Status	Text Box	Y
EcoVadis Sharing Status	EV_Sharing_Status	EVSharingStatus	Character	POZ_SUPPLIER_S_ATTRIBUT18	60 Character	EcoVadis Sharing Status	Text Box	Y
EcoVadis Request Outcome	EV_Request_Outcome	EVRequestOutcome	Character	POZ_SUPPLIER_S_ATTRIBUT19	60 Character	EcoVadis Request Outcome	Text Box	Y

Name	Code	API Name	Data Type	Table Column	Value Set	Prompt	Display Type	Readonly
Score Expired	EV_Score_Expired	EVScore_Expired	Character	POZ_SUPPLIER_ATTRIBUTES.ATTRIBUTE20	10 Characters	EcoVadis Score Expired	Text Box	Y
Global Score	EV_Global_Score	EVGlobalScore	Number	POZ_SUPPLIER.S.ATTRIBUTE_NUMBER6	3 Digits	EcoVadis Global Score	Text Box	Y
Environment Score	EV_Env_Score	EVEnvScore	Number	POZ_SUPPLIER.S.ATTRIBUTE_NUMBER7	3 Digits	EcoVadis Environment Score	Text Box	Y
Labor & Human Rights Score	EV_Lab_Score	EVLabScore	Number	POZ_SUPPLIER.S.ATTRIBUTE_NUMBER8	3 Digits	EcoVadis Labor & Human Rights Score	Text Box	Y
Ethics Score	EV_Fbp_Score	EVFbpScore	Number	POZ_SUPPLIER.S.ATTRIBUTE_NUMBER9	3 Digits	EcoVadis Ethics Score	Text Box	Y
Sustainable Procurement Score	EV_Sup_Score	EVSupScore	Number	POZ_SUPPLIER.S.ATTRIBUTE_NUMBER10	3 Digits	EcoVadis Sustainable Procurement Score	Text Box	Y
Score Published Date	EV_Score_Published_Date	EVScore_Published Date	Date	POZ_SUPPLIER.S.ATTRIBUTE_DATE9	FND_DATE_STANDARD	EcoVadis Score Published Date	Text Box	Y
Sharing Acceptance Date	EV_Sharing_Accepted_Date	EVSharingAccepted Date	Date	POZ_SUPPLIER.S.ATTRIBUTE_DATE10	FND_DATE_STANDARD	EcoVadis Sharing Acceptance Date	Text Box	Y

- e. Save and deploy the flexfields. For more information on setting up descriptive flexfields, see [Overview of Descriptive Flexfields](#).
6. Use page composer to make phone required during supplier registration. Phone number for supplier contact is required in the request sent to EcoVadis (see [Modify UI Components Using Page Composer](#)). If you have enabled **Redwood: Next Generation User Experience for Supplier Registration** feature, refer to the [What's New](#) on how to make phone required in the registration flow.

7. Opt-in to feature **Enable Outbound Supplier Profile Integration Using Oracle Integration Cloud**.
 - a. From the navigator, go to **My Enterprise** and then **Offerings**.
 - b. Select **Procurement** and click **Opt-in Features**.
 - c. Click the Features pencil icon for **Suppliers**.
 - d. For feature **Enable Outbound Supplier Profile Integration Using Oracle Integration Cloud**, select the **Enable** checkbox.

For more information on this Opt-in feature, see [Outbound Supplier Profile Integration Using Oracle Integration Cloud](#).

For more information on how to configure offerings, see [Configure Offerings](#).

Configure EcoVadis

To access EcoVadis from Oracle Integration and synchronize ESG scores, you'll need to have a valid EcoVadis subscription and an account to access EcoVadis APIs.

For more information on these configuration tasks, see [EcoVadis Help Center](#).

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the ERP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.ocs.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on the Oracle Procurement Cloud instance. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on the Oracle Procurement Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the ERP REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter the connection URL. For example: <code>https://your_domain_name.fa.ocs.oraclecloud.com</code>

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
User Name	Enter the user name of the account created for Oracle Integration on the Oracle Procurement Cloud instance. See Before You Install the Recipe .
Password	Enter the password of the account created for Oracle Integration on the Oracle Procurement Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the EcoVadis Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Specify the URL of your EcoVadis instance. For example: <code>https://api-sandbox.ecovadis-survey.com</code>

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Custom Two Legged Flow .
Access Token Request	-X POST -d grant_type=password&username=xxxxx&password=xxxxx <code>https://api-sandbox.ecovadis-survey.com/EVToken</code>

4. Under Optional security, enter the following details:

Field	Information to Enter
Refresh Token Request	-X POST -d grant_type=password&username=xxxxx&password=xxxxx https://api-sandbox.ecovadis-survey.com/EVToken
\$access_token	access_token
\$refresh_token	access_token
\$expiry	expires_in
\$token_type	token_type
access_token_usage	-H Authorization: \${token_type} \${access_token}

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the field names and corresponding values in the lookup table of the recipe.

- In the Lookups section, click the lookup name **EcoVadis**.
- Update the fields and corresponding values in the lookup table.

EcoVadis Field	Value
buyer_contact_first_name	Specify the first name of the internal user in your buying organization who will be responsible for communication with EcoVadis. A valid value is required.
buyer_contact_last_name	Specify the last name of the internal user in your buying organization who will be responsible for communication with EcoVadis. A valid value is required.
buyer_contact_email	Specify the email address of the internal user in your buying organization who will be responsible for communication with EcoVadis. A valid value is required. Do not use mailing list email.
buyer_phone	Specify the phone number of your buying organization that can be used for communication with EcoVadis. A value is optional.
campaign_type	Enter STD for long standard assessment process (12- 14 weeks) with EcoVadis. Enter RFP for short assessment process (Request For Proposal 4-6 weeks) with EcoVadis. A value is required.
data_last_update_in_days	Enter integer value larger than zero (0). Use this to specify the number of days in the past that the integration would look for ESG data update. For example, when set as 1, the integration will look for ESG data updated in the past one day. A value is required.

- Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle Procurement Cloud instance. Submit and approve a supplier registration. See [Supplier Registration Process](#).

This triggers the **Supplier EcoVadis Request ESG Score** integration. On successful execution of the integration flow, a request for ESG data will be sent to EcoVadis.
 - b. In the Integrations section of the project workspace, click **Actions**  on the **Supplier EcoVadis Sync ESG Score** integration flow, then select **Run**.
 - c. On the Configure and run page, click **Run**.

You've now successfully submitted the **Supplier EcoVadis Sync ESG Score** 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](#). It is recommended to run this integration daily.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to Oracle Procurement Cloud and navigate to the Suppliers work area. Check if the EcoVadis data is updated onto the supplier profile.

Related Documentation

- [Using the Oracle ERP Cloud Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)
- [EcoVadis Help Center](#)
- [Oracle Procurement Cloud - Suppliers Overview](#)
- [Oracle Procurement Cloud - Suppliers REST API](#)

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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)

- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe supports bi-directional data synchronization of incidents and resources between Oracle Field Service Cloud and Oracle Fusion Cloud B2C Service.

To use the recipe, you must install the recipe and configure the connections within the recipe. When an incident is created, updated or deleted in Oracle B2C Service, the appropriate integration flow of the recipe gets triggered and a corresponding activity is created, updated or deleted in Oracle Field Service Cloud. Similarly, when a resource is created or updated in Oracle Field Service Cloud, the appropriate integration flow of the recipe gets triggered and the corresponding resource is created or updated in Oracle B2C Service. Also, when an activity is updated in Oracle Field Service Cloud, the appropriate integration flow of the recipe gets triggered and the corresponding incident is updated in Oracle B2C Service.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Oracle Field Service Cloud, Version 17.2 or higher
- An account on Oracle Field Service Cloud with the Administrator role
- Oracle B2C Service, Version 16.8 or higher
- An account on Oracle B2C Service with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle B2C Service and Oracle Field Service Cloud instances in order to successfully connect to these external systems using Oracle Integration and synchronize incidents and resources.

Configure Oracle B2C Service

To access Oracle B2C Service using Oracle Integration and synchronize incidents and resources, you must perform certain configuration tasks on your Oracle B2C Service instance.

For general configuration tasks, see Prerequisites for Creating a Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.

In addition, you have to create a custom incident workspace and custom objects.

To find more information on workspaces, see:

- [Overview of Workspaces](#)
- [Create a Workspace](#)
- [Custom Workspaces](#)

Perform the following configuration tasks on your Oracle B2C Service instance.

1. Log in to the Oracle Service Cloud console.
2. In the navigation menu, select **Configuration**, click **Site Configuration**, and then click **Configuration Settings**.

3. In the **Key** field of the Configuration Settings window, enter EVENT_NOTIFICATION% and click **Search**.
4. Set values for the following configuration verbs and then save the changes.
 - EVENT_NOTIFICATION_SUBSCRIBER_USERNAME (for example, weblogic)
 - EVENT_NOTIFICATION_SUBSCRIBER_PASSWD (for example, welcome1)
 - EVENT_NOTIFICATION_MAPI_USERNAME (for example, icsuser)
 - EVENT_NOTIFICATION_MAPI_PASSWD (for example, test123)
 - EVENT_NOTIFICATION_SUBSCRIBER_ENABLE (set the value to YES)
5. Create required custom objects.
 - a. In the navigation menu, select **Configuration**, click **Database**, and then click **Object Designer**.
 - b. In the Object Designer window, select **CO** from the side menu.
 - c. Click **New** in the ribbon, and then select **Menu Only Object** from the drop-down menu.
 - d. In the Object Summary - Edit window, enter the object name (TimeSlot) in the **Name** field, and click **Add** under **Menu Object Options**.
 - e. Click **Save**.
 - f. Repeat steps a to e to create another custom object (ReminderTime).
6. Bind the newly created TimeSlot and ReminderTime custom objects to the incident workspace.
 - a. In the Object Designer window, expand the **CO** from the side menu and select **Incident**.
 - b. In the Objects Fields - Edit window, click the **Add New Field** button, and then select **Menu**.
 - c. In Field Details window, enter the name in the **Name** field.
 - d. In the **Menu** field, select the required menu object (C0.TimeSlot2) from the drop-down list.
 - e. Save and deploy the object designer.
 - f. Repeat steps a to e to add the C0.ReminderTime menu object.

Configure Oracle Field Service Cloud

To connect to Oracle Field Service Cloud from Oracle Integration, you must perform certain configurations tasks. See Prerequisites for Creating a Connection in *Using the Oracle Field Service Cloud Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle B2C Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the WSDL to use. See Prerequisites for Creating a Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username to connect to the Oracle B2C Service instance. See Prerequisites for Creating a Connection in <i>Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3</i> .
Password	Enter the password to connect to the Oracle B2C Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Field Service Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Field Service API URL	Enter the URL used to connect to your application and process requests. For production, the URL is <code>https://api.etadirect.com</code> . For development and testing, use the URL provided by Oracle when your Oracle Field Service Cloud instance is provisioned.
Instance ID	Enter the instance ID. The instance ID is provided by Oracle when your Oracle Field Service Cloud instance is provisioned.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication or Client Credentials .
Username and Password	If you selected Basic Authentication as security policy, then enter the username and password to access the destination web service. See Configure Oracle Field Service Cloud .

Field	Information to Enter
Client Id and Client secret	If you selected Client Credentials as security policy, then enter the client Id and password that you obtained. See Configure Oracle Field Service Cloud .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle B2C Service instance and create, update or delete an incident.
 - When an incident is created in Oracle B2C Service, the **Oracle OSvC OFSC Activity Create** integration flow is triggered, and it creates a corresponding activity in Oracle Field Service Cloud.
 - When an incident is updated in Oracle B2C Service, the **Oracle OSvC OFSC Activity Update** is triggered, and it updates the corresponding activity in Oracle Field Service Cloud.
 - When an incident is deleted in Oracle B2C Service, the **Oracle OSvC OFSC Activity Delete** integration flow is triggered, and it deletes the corresponding activity in Oracle Field Service Cloud.
 - b. Log in to your Oracle Field Service Cloud instance and create/update a resource or update an activity.
 - When a resource is created in Oracle Field Service Cloud, the **Oracle OFSC OSvC Resource Create** integration flow is triggered, and it creates a corresponding resource in Oracle B2C Service.
 - When a resource is updated in Oracle Field Service Cloud, the **Oracle OFSC OSvC Resource Update** integration flow is triggered, and it updates the corresponding resource in Oracle B2C Service.
 - When an activity is updated in Oracle Field Service Cloud, the **Oracle OFSC OSvC Incident Update** is triggered, and it updates the corresponding incident in Oracle B2C Service.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. If you've created, updated or deleted an incident in Oracle B2C Service, check if the corresponding activity has been created, updated or deleted in Oracle Field Service Cloud.
5. If you've created or updated a resource in Oracle Field Service Cloud, check if the corresponding resource has been created or updated in Oracle B2C Service. If you've updated an activity in Oracle Field Service Cloud, check if the corresponding incident has been updated in Oracle B2C Service.

Related Documentation

- *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
- *Using the Oracle Field Service Cloud Adapter with Oracle Integration 3*

Sync 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.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes inventory item records from Oracle Product Hub Cloud to Oracle E-Business Suite in real time.

To use the recipe, you must install the recipe and configure the connections and other resources within it. 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.

Note

- 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

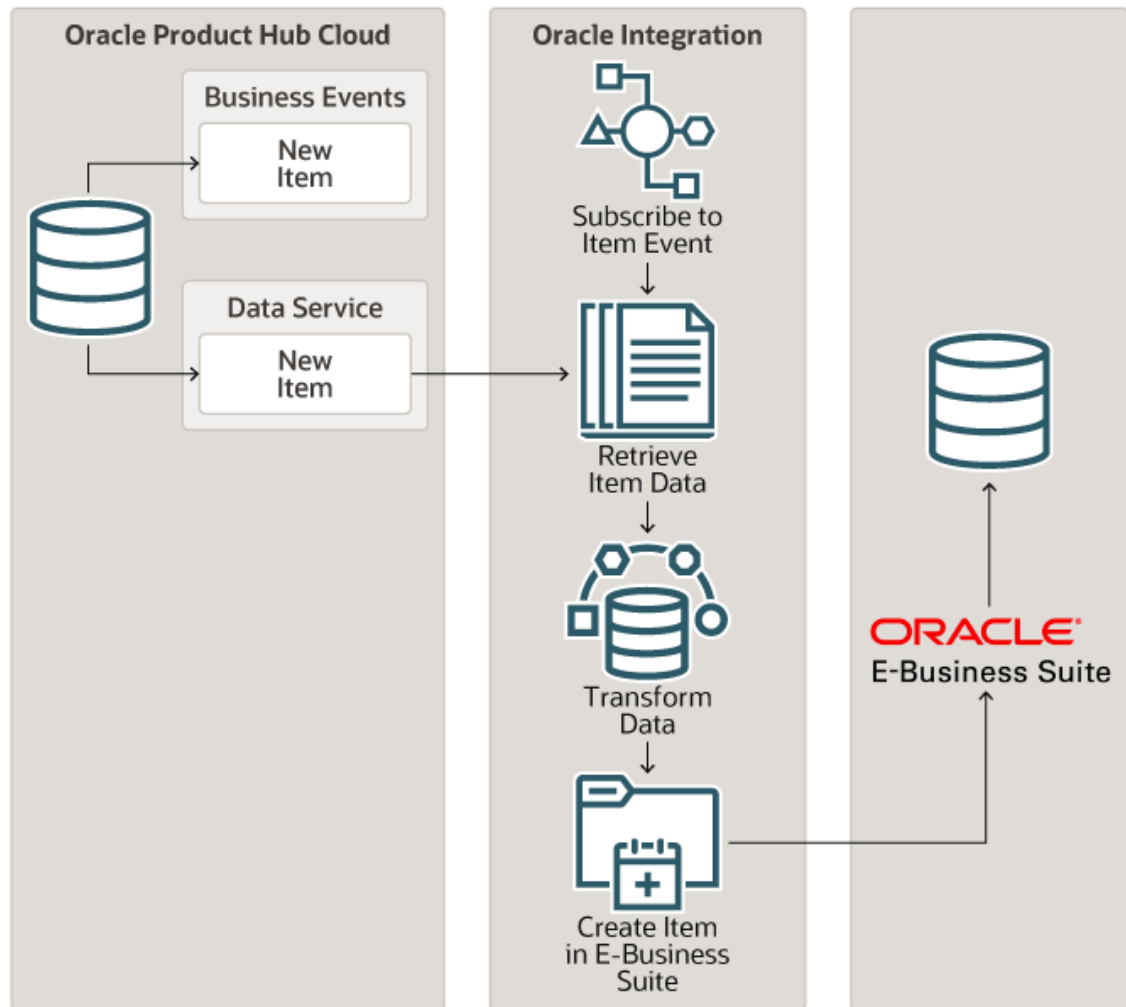
- Oracle Integration, Version 21.2.2.0.0 (210505.1400.40944) or higher
- Oracle E-Business Suite, Version 12.2.X
- Oracle ERP Cloud
- An account on Oracle E-Business Suite with the Administrator role
- Accounts on Oracle ERP Cloud with the Administrator role and the Product Data Steward role (in order 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.



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.

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.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this account to connect to Oracle ERP Cloud from Oracle Integration.

2. Assign the following roles to the user account. See *Assign Required Roles to an Integration User* in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user

Configure Oracle E-Business Suite

To access Oracle E-Business Suite from Oracle Integration, you must perform certain configurations on your Oracle E-Business Suite instance. In addition, you'll require a separate user account on Oracle E-Business Suite with necessary security grants.

Log in to your Oracle E-Business Suite instance as an **Administrator** and execute the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. 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* in *Using the Oracle E-Business Suite Adapter with Oracle Integration 3*. 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 `p_lmsvc` 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 to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle E-Business Suite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the URL of your Oracle E-Business Suite instance.
3. In the Security section, enter the following details:

Field	Information to Enter
Username	Enter the username of the account created for Oracle Integration on Oracle E-Business Suite.
Password	Enter the password of the account created for Oracle Integration on Oracle E-Business Suite.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and enter appropriate values.

1. In the Lookups section, click the lookup name.
2. Edit the lookup keys and enter required values.

Lookup key	Description
toEmail	Enter the email address to which run-time exception emails are to be sent.
fromEmail	Enter the email address from which run-time exception emails are to be sent.
p_organization_code	Enter the code of the organization in Oracle E-Business Suite with which to associate the new items created by this recipe.

3. Click **Save**. If prompted, click **Save** again.

4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to 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**.
 - c. 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 running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
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 3](#)

- *Using the Oracle E-Business Suite Adapter with Oracle Integration 3*

Sync Jira Issues with GitHub

Use this recipe to synchronize issues created in Jira with GitHub.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to use Oracle Integration to create or update corresponding issues in GitHub for issues created or updated in Jira.

To use the recipe, you must install the recipe and configure the connections and other resources within it. When an issue is created or updated in Jira, the integration flow gets triggered and it creates or updates a corresponding issue in GitHub. Basic data such as Self, Id, Fields and so on are mapped between Jira and GitHub platforms.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- Jira
- An account on Jira with the Administrator role
- GitHub
- An account on GitHub with the Administrator role

Before You Install the Recipe

You must perform certain configuration tasks on your Jira instance and GitHub instance in order to connect to these external systems using Oracle Integration, and successfully synchronize issues created or updated in Jira with GitHub.

Configure Jira

To configure Jira, see Prerequisites for Creating a Connection in *Using the Jira Adapter with Oracle Integration 3*.

Configure GitHub

To configure GitHub, see Prerequisites for Creating a Connection in *Using the GitHub Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Jira Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Jira host name. For example: `https://id.atlassian.net/`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Authorization Code Credentials selected.
Client Id	Enter the client id. See Configure Jira .
Client Secret	Enter the client secret.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the GitHub Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Authorization Code Credentials selected.
Client Id	Enter the client id. See Configure GitHub .
Client Secret	Enter the client secret.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Update Integration Properties

You can pass specific values while running the **Oracle Jira GitHub Create Issue Sync** integration by updating the integration property values.

1. In the Integrations section, click **Actions** . . . on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **Owner_name**: This integration property holds the name of the owner.
 - b. **Retry_count**: This integration property holds the value for the number of retries when an invoke fails. The default value is 3.
 - c. **emailTo**: This integration property holds the email id to which error notifications are sent.

3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Get the integration's endpoint URL.
 - a. In the Integrations section of the project workspace, click **Actions** . . . on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Endpoint metadata**.
 - c. In the panel that opens, copy the **Endpoint URL** value. This is the integration flow's endpoint URL.
3. Update the integration's endpoint URL in your Jira instance.

To register the endpoint URL:

- a. From your service project, select **Project settings**, then **Details**.
- b. In the **URL** field, enter the integration's endpoint URL that you obtained in step 2.
- c. Click **Save**.
4. Create an issue in Jira.
 - a. Select **Create**.
 - b. In the **Summary** field, enter the summary of the issue.
 - c. Complete all the required fields and any other fields.
 - i. *Optional*: To change which fields appear when you create another issue, select **Configure Fields** and select and deselect the fields you want. When you next create an issue, the selected fields will be displayed.
 - ii. *Optional*: To create other issues with the same **Project** and **Issue Type**, select **Create another** . Depending on your configuration and issue creation history, some fields may be pre-populated. Review these before creating the new issue.
 - d. Select **Create**.

By creating the new Jira issue, you've successfully triggered the **Oracle Jira GitHub Create Issue Sync** integration flow. The integration now creates a corresponding issue in GitHub.

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

6. Check if a corresponding new issue is created in GitHub.
 - a. Log in to your GitHub instance.
 - b. Select the repository.
 - c. At the top of the page, click **Issues**.

View the issues and verify that a new issue corresponding to the Jira issue is present.

Related Documentation

- *Using the Jira Adapter with Oracle Integration 3*
- *Using the GitHub Adapter with Oracle Integration 3*

Sync Leads Between Marketo and Salesforce

Use this recipe to synchronize leads between Marketo and Salesforce.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is a bidirectional recipe. It creates a corresponding lead record in Marketo when a lead record is set to *Open* status in Salesforce. In addition, it creates corresponding lead records in Salesforce for Marketo leads.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe.

When a lead record's status is set to *Open* in Salesforce, the app-driven integration flow (**Oracle SFDC Marketo Lead2Lead Create**) of the recipe is triggered, and it creates a corresponding lead record in Marketo. Further, you can run the scheduled integration flow (**Oracle Marketo Salesforce Lead2Lead CreateUpdate**) of the recipe manually or specify an execution schedule for it. When triggered, this integration flow fetches the lead records from the Marketo instance and, in turn, creates or updates corresponding leads in your Salesforce instance. Basic data, such as company, country, phone number, lead status, and so on, are synchronized between the two platforms.

System and Access Requirements

- Oracle Integration, Version 22.1.2.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Marketo
- An account on Marketo with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Marketo instances in order to successfully connect to these external systems using Oracle Integration and synchronize leads between them.

Configure Salesforce

To access Salesforce from Oracle Integration and create/read lead records, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the **Oracle SFDC Marketo Lead2Lead Create** integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce lead's status is set to *Open*. Finally, create a custom field for Salesforce lead records to store Marketo lead IDs.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
 - a. On the Salesforce Setup page:
 - i. Expand **Manage Users** under the Administer section in the left navigation pane.
 - ii. Click **Profiles**.
 - b. On the Profiles page, click **New Profile**.
 - c. On the resulting page:
 - i. Select **Standard User** in the **Existing Profile** field.
 - ii. Enter a name for the new profile, for example, API Enabled, and click **Save**.
The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the Accounts row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.

- In the Contacts row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the Price Books row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the Products row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the Leads row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, **Delete**, **ViewAll** and **Modify All** check boxes.
- iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the API Enabled profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
- i. Enter a first name and last name for the user, for example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field.
Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.
- Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
- i. Stay signed in as the new user and click the user name at the top of the page to open a menu.

- ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account.
Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
 3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version.
 4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization.
 5. On your Salesforce instance, create an outbound message to trigger the **Oracle SFDC Marketo Lead2Lead Create** integration flow of this recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.
If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.
 - c. On the New Outbound Message page:
 - i. Select **Lead** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the message.
 - In the **Endpoint URL** field, enter the following URL:

`https://<OIC_FQDN:OIC_SSL_PORT>/ic/ws/integration/v2/flows/salesforce/project/ORCL-R-SFDC_MARKT_SYNC_LEADS/ORCL-R-SFDC_LEA_SYNC_TO_MKT_LEA/1.0`

Where OIC_FQDN and OIC_SSL_PORT are the fully-qualified domain name and port of your Oracle Integration instance (for example, oicinstance.example.com:443).
- The new outbound message is created and displayed on your browser.
6. Now, create a workflow rule to send the outbound message (created previously) when a Salesforce lead's status is set to *Open*.
 - a. On the Salesforce Setup page:

- i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Workflow Rules**.
If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Rule** in the resulting page.
 - c. On the New Workflow Rule page:
 - i. Select **Lead** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:
 - Enter the rule's name.
 - Optionally, enter a description for the rule.
 - In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
 - In the Rule Criteria section, select the **formula evaluates to true** option, and define the following rule criteria:

(Lead: Lead Status EQUALS Open - Not Contacted) AND (Lead: Company NOT EQUAL TO null)
 - Click **Save & Next**.
 - iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
 - iv. On the Select Existing Actions page:
 - Select **Outbound Message** in the **Search** field.
 - In the **Available Actions** box, select the outbound message you created previously, for example, Outbound Message: <name of your outbound message>.
 - Click **Add** to move your selection to the **Selected Actions** box.
 - Click **Save**.
 - v. Click **Done**.
The new workflow rule is created and displayed on your browser.
 - d. Click **Activate** to activate the rule.
7. Create a custom field for lead records.
- This recipe uses unique IDs associated with lead records in Marketo to synchronize Marketo leads with Salesforce leads. Create a custom field for Salesforce lead records to hold the Marketo IDs.
- a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Customize**, then **Leads**, and then click **Fields**.
 - b. On the Lead Fields page, scroll to the Lead Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:

- Enter Lead Extension ID as the field label. Note that the field name is automatically populated based on the label you enter.
 - Enter 50 as the length.
 - Optionally, enter a description for the new field.
 - Select the **External ID** check box.
 - Click **Next**.
- iii. On the Establish field-level security page, select the **Visible** check box in the header row to grant edit access to the new field for all profiles. Click **Next**.
- iv. Leave all the layout check boxes selected and click **Save**.

On the Lead Fields page, you can see the new field added under the Lead Custom Fields & Relationships section.

Configure Marketo

To access Marketo from Oracle Integration and create/read lead records, you'll need to perform the following configuration tasks on your Marketo instance.

1. Create an API-only user role.
2. Create a customer service.
3. Obtain the Client ID and Client Secret.
4. Obtain the Munchkin ID.

For detailed information on the above steps, see Prerequisites for Creating a Connection in *Using the Marketo Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.

Field	Information to Enter
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce.

 Note

To the password, you must also append the security token generated for the same account.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle Marketo Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the Munchkin ID that you obtained earlier. See [Configure Marketo](#).
- In the Security section, enter the following details:

Field	Information to Enter
Client Id	Enter the client ID obtained earlier. See Configure Marketo .
Client Secret	Enter the client secret obtained earlier.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Lookup Table

The **ORACLE-BRT-MKT_SFDC_LOOKUP** table contains a sample mapping of lead-status values. Edit the table to map the lead-status values in your Marketo instance to that of your Salesforce instance.

- In the Lookups section, click the lookup name.
- Enter the values for the following keys.

Key	Description
Marketo_Value	Enter the lead-status values available in your Marketo instance (in separate rows). For example, Open, Qualified, and so on.
Salesforce_Value	Enter a corresponding lead-status value available in your Salesforce instance against each Marketo value.

- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration property for the **Oracle SFDC Marketo Lead2Lead Create** integration flow.

- In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
- In the Update property values panel, update the following integration property.
 - EmailTo:** This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
- Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

- Activate the recipe. See [Activate a Recipe or Accelerator](#).
- Run the recipe to create a Marketo lead when a Salesforce lead's status is set to *Open*.
 - Log in to your Salesforce instance.
 - On the Salesforce Setup page (in the Classic UI), click the **Leads** tab.
 - On the Leads Home page, in the **View** field, select an option to list the lead records, and click **Go!** if necessary.
The leads on your Salesforce instance are displayed.
 - Search for a lead record that is not in the *Open* state, and click its name to view it.
 - On the lead's page, click **Edit**.
 - On the Lead Edit page:
 - In the **Lead Status** field, select **Open - Not Contacted**.
 - Click **Save**.

You've now successfully triggered the recipe, which creates a corresponding lead record in Marketo.

- Run the recipe to create leads in Salesforce for Marketo lead records.
 - In the Integrations section of the project workspace, click **Actions**  on the **Oracle Marketo Salesforce Lead2Lead CreateUpdate** integration flow, then select **Run**.

The Configure and run page is displayed with the following parameter. Update the parameter with appropriate value.

- **LastRun:** This parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Marketo lead records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the lead records from the beginning, set the value as zero.

- a. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Marketo instance and check for the new leads created.
 - a. On the landing page, click the **My Marketo** tab.
 - b. In the left navigation pane, click **All People (APA)**.
 - c. Under the **People** tab, search for the required lead records.

You can search for the leads by their names, IDs, or any other configured fields.

6. Log in to your Salesforce instance and check for the new leads created.
 - a. On the Salesforce Setup page (in the Classic UI), click the **Leads** tab.
 - b. On the Leads Home page, in the **View** field, select an option to list the lead records, and click **Go!** if necessary.

The list of leads is displayed, which contains the records imported from Marketo.

- c. To view a lead record, click its name.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Marketo Adapter with Oracle Integration 3*

Sync New Cases from Microsoft Dynamics 365 to ServiceNow

Use this recipe to synchronize new cases from Microsoft Dynamics 365 to ServiceNow.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a new incident in ServiceNow when a new case event is created in Microsoft Dynamics 365.

To use this recipe, you must install the recipe and configure the connections and other resources within the recipe.

When a Microsoft Dynamics 365 case is created, the **Oracle MSD Create Case** integration flow of the recipe is triggered, and it creates a corresponding incident in ServiceNow by invoking the **Oracle MSD ServiceNow Case Sync** integration flow of the recipe. Basic data such as Case Title, Incident ID, System ID are synchronized between the Microsoft Dynamics 365 and ServiceNow platforms.

System and Access Requirements

- Oracle Integration 3
- Microsoft Dynamics 365
- An account on Microsoft Dynamics 365 with the Administrator role
- ServiceNow
- An account on ServiceNow with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Microsoft Dynamics 365 and ServiceNow to successfully connect to these external systems using Oracle Integration.

Configure Microsoft Dynamics 365

To configure Microsoft Dynamics 365, see *Prerequisites for Creating a Connection in Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3*.

Configure ServiceNow

You can either use admin account credentials or custom account credentials while configuring the ServiceNow connection in Oracle Integration.

If you do not want to use an admin account, create a custom integration user and assign the required role and permissions to the user. For information about creating a custom user, see *Prerequisites for Creating a Connection in Using the ServiceNow Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Microsoft Dynamics 365 Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Microsoft Dynamics 365 Organization URL	Enter the Microsoft Dynamics 365 organization URL.
Tenant ID	Enter the tenant id. See Configure Microsoft Dynamics 365 .
API Version	Enter the API version. See Configure Microsoft Dynamics 365 .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Client Credentials .
Client Id	Enter the client id. See Configure Microsoft Dynamics 365 .
Client Secret	Enter the client secret. See Configure Microsoft Dynamics 365 .

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the ServiceNow Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the ServiceNow instance name. For example: `https://host_name.servicenow.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the ServiceNow username.
Password	Enter the ServiceNow password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties with appropriate values.

Note

This is an application driven integration, and you can update the property values only before activating the recipe.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Integrations section, click **Actions** ... on the **Oracle MSD ServiceNow Case Sync** integration flow, then select **Update property values**.
3. In the Update property values panel, update the integration properties with appropriate values.
 - a. **emailTo**: This integration property holds the email address to which runtime exception emails are sent. Enter an email address of your choice.
 - b. **retry_count**: This integration property holds the number of retries to attempt if a record fails while importing employee records at run time. Enter a number of your choice.
4. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must create a new case in the Microsoft Dynamics 365 application.

- a. Sign in to the Power Pages site and select **Support**.
- b. A list of cases logged by you is displayed.
- c. To create a new case, select **New Case**.
- d. Enter the required details in the form, such as **Title**, **Customer**, **Case Type**, **Description**, and so on.
- e. Select **Submit**.

By default, the following error appears if you choose a contact that doesn't belong to the selected account while creating a case: *The specified contact doesn't belong to the account selected as the customer. Specify a contact that belongs to the selected account and then try again.*

You can customize this default error by adding a [Content Snippet](#) with the Name as **cases/unrelatedcontact** and Value as your custom error message text.

Note

Create a custom column named **CrossRef_Servicenow_IncidentId** to store the ID of the newly created incident in ServiceNow. See [Create a column](#).

You've now successfully submitted the integration for execution.

When you create a new case in Microsoft Dynamics 365, the **Oracle MSD Create Case** integration flow gets triggered, and a corresponding incident is created in the ServiceNow instance.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your ServiceNow instance and check if a new company has been created.
 - a. In the left navigation pane, under **Service Desk**, click **Incidents**.
 - b. On the Incidents page, search for the newly created incident with a short description.

Check if the newly created incident data has been updated under the **CrossRef_Servicenow_IncidentId** field in the Microsoft Dynamics 365 case that you created earlier.

Related Documentation

- *Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3*
- *Using the ServiceNow Adapter with Oracle Integration 3*

Sync Oracle NetSuite Customers with Oracle CX Sales and B2B Service Accounts

Use this recipe to synchronize Oracle NetSuite customers with Oracle CX Sales and B2B Service accounts.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates or updates a corresponding Oracle CX Sales and B2B Service account when a customer is created or updated in Oracle NetSuite. Additionally, whenever an account is created or updated in Oracle CX Sales and B2B Service, the recipe creates or updates a corresponding customer in Oracle NetSuite.

To use the recipe, you must install the recipe and configure the connection and other resources within it. The recipe contains three integration flows - **Oracle NetSuite CX Sales Customer Upsert**, **Oracle CX Sales NetSuite Customer Create** and **Oracle CX Sales NetSuite Customer Update**.

- When a new customer is created or a customer record is updated in Oracle NetSuite, the user event script associated with the customer gets triggered in NetSuite. The user event script provides the JSON data of the customer and subsequently triggers the **Oracle**

NetSuite CX Sales Customer Upsert integration flow in Oracle Integration. The integration then creates or updates a corresponding account record in Oracle CX Sales and B2B Service.

- When a new account is created in Oracle CX Sales and B2B Service, the **Oracle CX Sales NetSuite Customer Create** integration flow gets triggered in Oracle Integration. The integration creates a corresponding customer record in Oracle NetSuite.
- When an account is updated in Oracle CX Sales and B2B Service, the **Oracle CX Sales NetSuite Customer Update** integration flow gets triggered in Oracle Integration. The integration then updates the corresponding customer record in Oracle NetSuite.

Basic data such as customer/account name, phone, email, company name and address are synchronized between Oracle NetSuite and Oracle CX Sales and B2B Service platforms.

System and Access Requirements

- Oracle Integration, Version 24.10 or higher
- Oracle NetSuite
- An account on Oracle NetSuite with the Administrator role
- Oracle CX Sales and B2B Service
- An account on Oracle CX Sales and B2B Service with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite and Oracle CX Sales and B2B Service instances in order to successfully connect to these external systems using Oracle Integration, and synchronize customers and accounts.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See [Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL](#).
2. Perform the TBA-related configuration tasks. See [Prerequisites for the Token-Based Authentication Security Policy](#).

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
List	Currency	Full
List	Customers	Full
List	Subsidiaries	Full
List	Contacts	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Create a custom field for customer record. See [Creating Custom Transaction Body Fields](#). While creating a custom field, enter the following values.
 - a. Specify the value for **Label** as CxSalesPartyNumber .
 - b. Specify the value for **ID** as id_cxsalespartynumber.
 - c. In the **Applies To** tab, select **Customer**.
 - d. Click **Save**.
4. Configure and deploy the User Event script.
 - a. Download the sample user event script: [NS-CXSales.js](#).
 - b. Open the script in a text editor such as Notepad++ and make the following updates:
 - i. On **line 30**, there's a variable with the value set to `***your admin user email***`. Replace this value with the email address of the integration user that you created. The end result should look like this: `var integrationUserEmail = "john@netsuite.com"`.
 - ii. On **line 85**, there's a variable with the value set to `*The base URL of your Oracle Integration instance*`. Replace this value with the domain name of your Oracle Integration instance. Make sure to omit everything after the .com top-level domain.
 - iii. On **line 83**, there's a variable with the value set to your username. Replace this value with the username of your Oracle Integration instance.
 - iv. On **line 84**, there's a variable with the value set to your password. Replace this value with the password of your Oracle Integration instance.
 - v. Save the changes made to the script.
 - c. In NetSuite, navigate to **Documents**, then **Files** and then **SuiteScripts**.
 - d. Within the SuiteScripts folder, create a new folder. For example: Stripe Integration.
 - e. Upload the script that you modified and saved in step b to the newly created folder.
 - f. Go to **Customization**, then **Scripting**, and then **Scripts**.
 - g. In the **Script File** field, start typing `Oracle_NS_Stripe_Customer_Create.js`, and select the auto-completed option.
 - h. Click **Create Script Record**.
 - i. Set the **Name** field to `Oracle NS Stripe Customer Script` and **ID** to `_oracle_stripe_cust`.
 - j. Click **Save** and then click **Deploy Script**.
 - k. Set the **Applies To** field to **Customer** and the **ID** to `_oracle_stripe_cust_depl`.
 - l. Set the **Status** to **Deployed**.
 - m. In the **Audience** tab, select all check boxes where applicable or select all items in the respective lists.
 - n. Click **Save**.

Configure Oracle CX Sales and B2B Service

To successfully connect to Oracle CX Sales and B2B Service using Oracle Integration, you must perform certain configurations on your Oracle CX Sales and B2B Service instance.

1. Create an integration user account.

To invoke an Oracle CX Sales and B2B Service service catalog or event catalog web service from , you create a separate user.

- a. Log in to the Oracle CX Sales and B2B Service with a user with system administrator privileges.
- b. Go to **Navigator > My Team > Manage Users**.
- c. In the Manage Users page, click **Manage Users**.
- d. Click the **Create New User** icon beside **Show Photo**.
- e. Enter the following information, and click **Save**.

Field	Description
Last Name	Enter FUSION_APPS_ICS_APPID
Email	Enter a valid email address.
Hire Date	Enter the date.
User Name	Enter FUSION_APPS_ICS_APPID.
Person Type	Enter Employee.
Legal Employer	Select a valid legal organization.
Business Unit	Select a valid business unit.
Send user name and password	Select this checkbox.

A notification email is sent to the email address after the user is created.

- f. Log out of the Oracle CX Sales and B2B Service.
 - g. Log in to the Oracle CX Sales and B2B Service instance with FUSION_APPS_ICS_APPID and the temporary password provided in the notification email.
 - h. Change the password when prompted at the first log in.
The Oracle CX Sales and B2B Service welcome page appears.
 - i. Log out of the Oracle CX Sales and B2B Service.
2. Assign integration roles.

Use the Oracle Security Console to assign the integration user with certain roles and privileges.

Note

Access to the Security Console is provided by the predefined **Security Manager** role.

- a. Select **Navigator > Tools > Security Console**.
- b. On the top right corner of the window, click **Create Role**.
- c. In the Create Role: Basic Information page, create a new record with the following information and click **Next**.

Parameter	Value
Role Name	OIC Integration Role
Role Code	INT_ OIC_ Integration_ Role

Parameter	Value
Role Category	CRM - Job Roles
Description	Custom Role for Accessing OSC Services Catalog

- d. In the Create Role: Functional Security Policies page, click **Add Functional Security Policy**.
- e. In the Add Function Security Policy page, enter `FND_MANAGE_CATALOG_SERVICE_PRIV` in the Search box and click **Add Privilege to Role**. Click **Next**.
- f. In the Create Role: Data Security Policies page, click **Next**.
- g. Add the **Sales Administrator** and the **SOA operator** roles, and then click **Next**.
To add the **Sales Administrator** role, do the following on the Create Role: Role Hierarchy page:
 - i. Click **Add role**.
 - ii. In the Search field, enter Sales Admin.
 - iii. Select the **Sales Administrator** role, and then click **Add Role Membership**.
 - iv. Close the Add Role Membership window.To add the **SOA Operator** role, do the following on the Create Role: Role Hierarchy page:
 - i. Click **Add role**.
 - ii. In the Search field, enter SOA Operator.
 - iii. Select the **SOA Operator** role, and then click **Add Role Membership**.
 - iv. Close the Add Role Membership window.
- h. Assign the integration user to the roles.
On the Create Role: Users page, click **Add user** and then do the following:
 - i. In the Search field, enter `FUSION_APPS_ICS_APPID`.
 - ii. Choose the **FUSION_APPS_ICS_APPID** user and then click **Add user to Role**.
 - iii. Close the Add user window.
- i. Click **Next**, review the details on the Summary and Impact page, and click **Save and Close**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Oracle NetSuite WSDL URL. For example: https://webservices.netsuite.com/wsd/v2022_1_0/netsuite.wsdl. See [Configure Oracle NetSuite](#).

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See Configure Oracle NetSuite .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite.
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle CX Sales and B2B Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: https://your_domain_name.fa.DC.oraclecloud.com.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Configure Oracle CX Sales and B2B Service .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORACLE-BRT-EC_NS_SETTINGS** lookup table.

1. In the Lookups section, click the lookup name.
2. Edit lookup keys in the **ECCode** and **NSCode** columns and map Oracle NetSuite values with Oracle CX Sales and B2B Service values.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle NetSuite instance and create a new customer or update a customer.

This triggers the **Oracle NetSuite CX Sales Customer Upsert** integration flow which subsequently creates or updates a corresponding account in Oracle CX Sales and B2B Service.
 - b. Log in to your Oracle CX Sales and B2B Service instance and create a new account.

This triggers the **Oracle CX Sales NetSuite Customer Create** integration flow which creates a corresponding customer in Oracle NetSuite.
 - c. Now, update an account in your Oracle CX Sales and B2B Service instance.

This triggers the **Oracle CX Sales NetSuite Customer Update** integration flow which updates a corresponding customer in Oracle NetSuite.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify if a corresponding account has been created or updated in your Oracle CX Sales and B2B Service instance for the new/updated Oracle NetSuite customer.
5. Similarly, verify if a corresponding customer has been created or updated in your Oracle NetSuite instance for the new/updated Oracle CX Sales and B2B Service account.

Related Documentation

- *Using the Oracle NetSuite Adapter with Oracle Integration 3*
- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*

Sync Oracle NetSuite Items with QuickBooks Products

Use this recipe to synchronize Oracle NetSuite items with products in QuickBooks.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes inventory and sales service items in Oracle NetSuite with products in QuickBooks, according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe package and configure the connections and other resources within the recipe. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

The recipe has the following scheduled integration flows:

- **Oracle NS ServiceItem QB Product Sync:** When triggered, the integration flow fetches the sales service items from Oracle NetSuite and creates corresponding products in QuickBooks.
- **Oracle NSInventoryItem QBProduct Sync:** When triggered, the integration flow fetches the inventory items from Oracle NetSuite and creates corresponding products in QuickBooks.

Subsequently, the recipe updates the Oracle NetSuite item record with the QuickBooks product ID in the Quickbook ProductItem Id field. If the recipe fails to do so, the corresponding QuickBooks product is marked as inactive. Basic data, such as account and unit price are synchronized between the two platforms.

Note

The recipe creates products in QuickBooks only for Oracle NetSuite item records in which the Quickbook ProductItem Id field is empty.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0 or higher
- Oracle NetSuite
- An account on Oracle NetSuite with the Administrator role
- QuickBooks
- An account on QuickBooks with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle NetSuite and QuickBooks instances in order to successfully connect to these external systems using Oracle Integration and synchronize items and products.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

In addition, you must create a NetSuite saved search. The search criterion specified in the saved search will be used to fetch the sales orders that are to be imported to Oracle ADW.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See [Enable Features on Oracle NetSuite](#) and [Assemble the Oracle NetSuite WSDL URL](#).
2. Perform the TBA-related configuration tasks. See [Prerequisites for the Token-Based Authentication Security Policy](#).

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
List	Currency	Full
List	Customers	Full
List	Subsidiaries	Full
List	Contacts	Full
Setup	User Access Tokens	Full
Setup	Log in using Access Tokens	Full
Setup	SOAP Web Services	Full

3. Configure saved search to get the inventory items that satisfy the search criterion. The items that are the output of the search result will be created as products in QuickBooks.
 - a. Navigate to the New Saved Search window using any one of the following navigation options:
 - On the NetSuite home page, select **Reports**, then choose **Saved Searches**, then **All Saved Searches**, and then **New**.
 - On the NetSuite home page, select **Transactions**, then choose **Management**, then **Saved Searches**, and then **New**.
 - b. On the New Saved Search window, select the object type to perform the search operation. For example, select **Item**.
 - c. Add a title and ID for the saved search in the SEARCH TITLE and ID fields respectively.

The title and ID for the saved search will help you in keeping track of the saved search when you want to reuse it to re-query latest data that matches the search criteria.
 - d. Set the search criteria conditions. Select the **Criteria** tab and specify the conditions. For example, you can specify a condition to filter items with the custom field **Quickbook ProductItem Id** as empty. If the **USE EXPRESSIONS** check box is selected, you can also use additional options such as AND, OR, NOT. Also, specify the type to be **Inventory**.
 - e. Configure the search results that are to be displayed. Select the **Results** tab, and in the **Columns** tab of the page, select the fields that you want to be displayed in the search result.

Similarly, create another saved search for sales service items. Specify a condition to filter items with the custom field **Quickbook ProductItem Id** as empty, type as **Service Item**, and the subtype as **Sales**.

Configure QuickBooks

To access the QuickBooks instance from Oracle Integration and synchronize items and products, you'll need to create a user account for Oracle Integration on QuickBooks and make

a few other configurations. See *Prerequisites for Creating a Connection in Using the QuickBooks Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle NetSuite Connection

Use the following steps to configure the **Oracle NetSuite** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, <code>https://webservices.netsuite.com/wsd/ <NetSuite_application_version>/netsuite.wsdl</code> , where <code><NetSuite_application_version></code> is the version of the NetSuite application. For example, <code>v2015_1_0</code> . See <i>Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See <i>Create an Integration Record for Oracle Integration in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See <i>Create an Access Token for the User Account in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Token Secret	Enter the token secret provided by Oracle NetSuite.
Account ID	Enter your Oracle NetSuite account identifier. See <i>Make a Note of the NetSuite Account ID in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

Note

You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.

5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle QuickBooks Connection

Use the following steps to configure the **Oracle QuickBooks** connection.

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
QuickBooks Environment	Select Production .

3. In the Security section, enter the following details:

Field	Information to Enter
Client id	Enter the Client ID obtained when you configured QuickBooks. See Configure QuickBooks .
Client secret	Enter the Client Secret obtained when you configured QuickBooks.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Provide Consent**.
6. In the resulting Sign in dialog, enter your Oracle Integration user name and password, and click **Sign in**.

You'll now be redirected to the QuickBooks sign in page.

7. Enter your QuickBooks account credentials.

A page is displayed asking for permission to interact with the account.

8. Click **Yes**.

You're informed that access is allowed. You can now switch back to the Oracle QuickBooks Connection window of Oracle Integration to test your connection.

9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

10. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **Account_lookup** lookup table to map the account names in Oracle NetSuite with QuickBooks account IDs.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

netsuite_account_name	quickbooks_account_id
1150 Employee Advances	57
Inventory Asset	81
Accounts Receivable	84
1090 Undeposited Funds	4
Uncategorized Income	30
1300 Prepaid Expenses	3
6010 Advertising	7
6020 Automobile Expense	55
4000 Sales	79
Cost of Goods Sold	80
6080 Equipment Rental	29
6100 Insurance Expense	11
6150 Office Expense	15

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - i. To synchronize sales service items, select **Oracle NS ServiceItem QB Product Sync**.
 - ii. To synchronize inventory items, select **Oracle NSInventoryItem QBProduct Sync**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. In your QuickBooks instance, check for new products.
 - a. Log in to your QuickBooks instance.
 - b. Click **Products and Services** in the homepage, and check for new products.

Related Documentation

- [Using the Oracle NetSuite Adapter with Oracle Integration 3](#)
- [Using the QuickBooks Adapter with Oracle Integration 3](#)

Sync Oracle Siebel Employees with Oracle Field Service Cloud Technicians

Use this recipe to synchronize Oracle Siebel employees with Oracle Field Service Cloud technicians.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates or updates a technician in Oracle Field Service Cloud when an employee is created or updated in Oracle Siebel.

To use the recipe, you must install the recipe and configure connections and other resources within it. When an employee is created or updated in Oracle Siebel, a Siebel workflow event is triggered. This in turn triggers the main integration flow of the recipe (**Oracle Siebel OFS Employee Sync**) which had subscribed to Siebel workflow events. Subsequently, the main integration flow of the recipe calls the helper integration flow (**Oracle Siebel OFS Employee Sync Helper**) which then creates or updates a corresponding technician in Oracle Field Service Cloud. Basic data such as name, address, contact details are synchronized between Oracle Siebel and Oracle Field Service Cloud.

System and Access Requirements

- Oracle Integration, Version 24.06 or higher
- Oracle Siebel
- Oracle Field Service Cloud
- An account on Oracle Siebel with the Administrator role
- An account on Oracle Field Service Cloud with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Siebel instance and Oracle Field Service Cloud instance in order to connect to these external systems using Oracle Integration and successfully synchronize Oracle Siebel employees with Oracle Field Service Cloud technicians.

Configure Oracle Siebel

Perform general configuration tasks for creating a connection. See Prerequisites for Creating a Connection in *Using the Oracle Siebel Adapter with Oracle Integration 3*.

Get the Swagger JSON file from Oracle Integration and perform the following tasks:

1. Log in to the Oracle Siebel instance.
2. Import the Swagger JSON file into Siebel via the Web Service Wizard (Siebel Web Tools).
3. Navigate to **Sitemap**, then **Administration - Web Services**, and then **Outbound REST Services** to verify the proxy business service.
4. Under **Siebel Objects** and then **Integration Objects**, verify any XML IO generated.
5. Create the input payload.
6. Provide security parameters such as user id, password and authentication type.
7. Call the proxy business service by providing the payload.

For more information, see the [Siebel Rest API guide](#).

Configure Oracle Field Service Cloud

Perform the general configuration tasks for creating a connection. See Prerequisites for Creating a Connection in *Using the Oracle Field Service Cloud Adapter with Oracle Integration 3*.

In addition, you have to create skills in Oracle Field Service Cloud. These skills must correspond to skills in Oracle Siebel. All skills available in Oracle Siebel are created with the pattern `ItemTypeName_SkillName`. For example, if the skill category is Language and skill name is english, you need to create `Language_en` in Oracle Field Service Cloud.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle SBL-OFS REST Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Oracle SBL-OFS OFSC Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Field Service Cloud API URL	Enter the URL of your Oracle Field Service Cloud instance.
Instance ID	Enter the instance ID.

3. In the Security section, enter the following details:

Field	Information to Enter
Security policy	Select Client Credentials .
Client Id	Enter the client Id.
Client Secret	Enter the client secret.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle SBL-OFS REST OFS Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL.
Connection URL	Enter the URL of your Oracle Field Service Cloud instance. For example: <code>https://<serviceinstanceid>.etadirect.com</code>

3. In the Security section, enter the following details:

Field	Information to Enter
Security policy	Select Basic Authentication .
Username	Enter the username of the account created for Oracle Integration on the Oracle Field Service Cloud instance. See Configure Oracle Field Service Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle Field Service Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Update Integration Properties

Update the integration properties for the **Oracle Siebel OFS Employee Sync Helper** integration flow.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.

2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **FromAddress**: This property holds the value of the email address from which error or failure notifications are sent. You can get the value to enter by navigating to **Settings**, then **Integrations**, and then **Notifications**.
 - b. **ToAddress**: This property holds the value of the email address to which error or failure notifications are sent. Enter an email of your choice.
 - c. **ParentResourceID**: This property holds the root organization English name. Get the value to enter by navigating to **Configuration** and then **Organizations**.
 - d. **BucketResourceLabel**: This property holds the label value corresponding to the bucket. Get the value to enter by navigating to **Configuration** and then **Resource Types**.
 - e. **BucketLanguageCode**: This property holds the language code of the bucket. For example, if the default language is English then enter en.
 - f. **TechnicianResourceLabel**: This property holds the label corresponding to technician. Get the value to enter by navigating to **Configuration** and then **Resource Types**.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

You have to update the integration properties of the **Oracle Siebel OFS Employee Sync** integration flow after activating the recipe. See [Activate and Run the Recipe](#).

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the following properties for the **Oracle Siebel OFS Employee Sync** integration flow.
 - **FromAddress**: This property holds the value of the email address from which error or failure notifications are sent. You can get the value to enter by navigating to **Settings**, then **Integrations**, and then **Notifications**.
 - **ToAddress**: This property holds the value of the email address to which error or failure notifications are sent. Enter an email of your choice.
3. Run the recipe.

Create an employee in Oracle Siebel. This triggers the main integration flow **Oracle Siebel OFS Employee Sync** to run which in turn calls the helper integration flow **Oracle Siebel OFS Employee Sync Helper**. The helper integration flow is submitted successfully for execution.
4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to Oracle Field Service Cloud and check if a corresponding technician has been created. Verify if basic data is synchronized between the Oracle Siebel employee and the Oracle Field Service Cloud technician.

Related Documentation

- *Using the Oracle Siebel Adapter with Oracle Integration 3*
- *Using the Oracle Field Service Cloud Adapter with Oracle Integration 3*

Sync Orders Between Oracle CX Sales and B2B Service and Oracle E-Business Suite

Use this recipe to synchronize orders between Oracle CX Sales and B2B Service and Oracle E-Business Suite.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes orders in Oracle CX Sales and B2B Service with Oracle E-Business Suite orders in real time. It uses the standard Oracle CX Sales and B2B Service 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 order is created in Oracle CX Sales and B2B Service, the integration flow of the recipe is triggered, and the recipe creates a corresponding order record in Oracle E-Business Suite. Basic data such as quantity, price, billing location, shipping location, email, and currency are synchronized between the Oracle CX Sales and B2B Service and Oracle E-Business Suite platforms.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Oracle CX Sales and B2B Service
- An account on Oracle CX Sales and B2B Service with the Administrator role
- Oracle E-Business Suite
- An account on Oracle E-Business Suite with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle CX Sales and B2B Service and Oracle E-Business Suite instances in order to successfully connect to these external systems using Oracle Integration and synchronize orders.

Configure Oracle CX Sales and B2B Service

To access the Oracle CX Sales and B2B Service instance from Oracle Integration, you'll require a separate user account on Oracle CX Sales and B2B Service.

Perform the following tasks to configure Oracle CX Sales and B2B Service for this recipe.

1. Create a new user account for Oracle Integration. You'll use the credentials of this user account to connect to Oracle CX Sales and B2B Service from Oracle Integration.

- For the procedure to create the integration user account, see *Creating an Integration User Account* in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
- Assign the following roles to the user account:
 - Integration Specialist
 - FND_MANAGE_CATALOG_SERVICE_PRIV

For steps to assign roles, see *Assign Integration Roles* in *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
 - Make a note of the user name and password you set for the account.

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](#).
 - 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.
 - Deploy the web services in your Oracle E-Business Suite instance.

Note

Ensure that you use an HTTPS URL while logging in to your Oracle E-Business Suite instance so that the generated WSDL also consists of HTTPS internal references.

- Navigate to **Integrated SOA Gateway**, and select **Integration Repository**.
- The following table lists the supported web services. For each web service listed below, perform the following steps if the web service has not been deployed.

Web Service	Internal Name	Type
Customer Account	HZ_CUST_ACCOUNT_V2PUB	PL / SQL
Customer Account Site	HZ_CUST_ACCOUNT_SITE_V2PUB	PL / SQL

Web Service	Internal Name	Type
Location	HZ_LOCATION_V2PUB	PL / SQL
Party Site	HZ_PARTY_SITE_V2PUB	PL / SQL
Sales Order Service	OE_INBOUND_INT	PL / SQL

- i. Click **Search**.
- ii. Enter the internal name and type of the web service, and then click **Go**.
- iii. In the search results, click the REST web service.
The details of the web service are displayed.
- iv. Click the name of the interface to view the details.
If the REST service is unavailable, enter the service alias and deploy the interface as a REST service.
- v. On the REST Web Service tab, click **Deploy**.
- vi. In the **Grants** area, select the service name, and then click **Create Grant**.
- vii. On the **Create Grants** page, enter the user's name in the format LastName, FirstName.
- viii. In the search results pop-up menu that appears, click the user's name.
The page refreshes itself.
- ix. Click **Apply**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle CX Sales and B2B Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: CX Sales and B2B Service Host.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Configure Oracle CX Sales and B2B Service .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle E-Business Suite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the URL of your Oracle E-Business Suite instance.
3. 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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured.

A message confirms if your test is successful.

6. To return to the Configuration Editor, click **Go back** .

Configure the Lookup Table

Edit the **EBS-CONFIG-LOOKUP** lookup table to configure the fields in Oracle E-Business Suite.

1. In the Lookups section, click the lookup name.
2. Enter the values for the following keys.

Key	Description
ToAddress	Specify the email address to which the error notification emails are sent. Enter an email address of your choice.

Key	Description
FromAddress	Specify the email address from which the error notification emails are sent. Enter an email address of your choice. See Troubleshoot Oracle Cloud Infrastructure Notification Email Configuration to Ensure Proper Delivery .
RespApplication	Specify the short name of the application. For example, PER.
SecurityGroup	Specify the security group key. For example, STANDARD.
NLSLanguage	Specify the NLS language. For example, AMERICAN.
Org_Id	Specify the organization ID. For example, 204.
OrderTypeId	Specify the organization type ID. For example, 1430.
AccountCreatedByModule	Specify the name of module from which the account was created. For example, ASO_CUSTOMER_DATA.
OrderResponsibilityName	Specify the order responsibility name. For example, ORDER_MGMT_SUPER_USER.
OrderRespApplication	Specify the short name of the order application. For example, ONT.
OrderApiVersionNumber	Specify the API version of Order Management. For example, 1.0.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe. To do so, create an order in Oracle CX Sales and B2B Service. See [Create Sales Orders](#).

You've now successfully triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if new orders are created in your Oracle E-Business Suite instance.
 - a. Log in to your Oracle E-Business Suite instance.
 - b. Open the **Order Organizer** window.
 - c. In the Quote/Order Information tab, specify the details of the order and click **Find**.
 - d. The **Find** window is displayed with the details of the required sales order.

Related Documentation

- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
- *Using the Oracle E-Business Suite Adapter with Oracle Integration 3*

Sync Orders Between Shopify and SAP S/4HANA

Use this recipe to synchronize orders between Shopify and SAP S/4HANA using OData adapter. The recipe also synchronizes newly created Shopify customers in SAP S/4HANA system.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a sales order in SAP S/4 HANA whenever an order is created in Shopify.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When an order is created in Shopify, the integration flow of the recipe is triggered, and it creates a corresponding sales order in the SAP S/4HANA instance. Basic data such as Customer Id, Name, Address, Quantity, and Price are synchronized between the Shopify and SAP S/4HANA platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Shopify
- An account on Shopify with the Administrator role
- SAP S/4 HANA
- An account on SAP S/4HANA with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Shopify and OData instances to successfully connect to these external systems using Oracle Integration and synchronize orders from Shopify to SAP S/4 HANA.

Configure Shopify

To configure Shopify, see Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Configure OData for SAP S/4HANA

To configure OData, see Prerequisites for Creating a Connection in *Using the OData Adapter with Oracle Integration 3*.

Create a custom field in SAP S/4 HANA to store Shopify customer id (Example: YY1_Shopify_customer1_bus).

To create a custom field in SAP S/4 HANA:

1. Log in to SAP S/4 HANA.
2. In the search field, search for **custom field** app.
3. Under the **Custom Fields** tab, to create a new custom field, click the plus icon.
The Business Context pop-up window appears.
4. In the search field, search for the Business Partner, then from the list, select **Master Data: Business Partner**.
A New Field pop-up appears.
5. In the New Field pop-up, provide the following details.

Attribute	Description
Label	Provide a name for the custom field.
Identifier	A default prefix YY1 followed by the Label (Custom field name) auto-populates.
Tooltip	Auto-populates the same value provided for the label.
Type	Data type

6. Click **Create and Edit**.
7. Under User Interfaces, to enable the OData Service usage, click **Enable Usage**.
8. Click **Save**, then **Publish**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Shopify Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Shopify Security Policy .
Username	Enter the username of the account created for Oracle Integration on Shopify. See Configure Shopify .
Password	Enter the password of the account created for Oracle Integration on Shopify.

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the OData Connection for S/4HANA

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
OData Service Root URL	Enter the OData Service Root URL. For example: <code>https://<host>/sap/opu/odata/sap/<service_name></code>
OData Version	Enter the OData Version. Example: V2

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of the account created for Oracle Integration on OData. See Configure OData for SAP S/4HANA .
Password	Enter the password of the account created for Oracle Integration on OData.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Update Integration Properties

Users can pass the value while running the integration by updating the integration property value.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **EmailTo**: This integration property holds the email address to which error notification emails are sent. Enter an email address of your choice.
 - b. **retryCount**: This integration property holds the number of retries when an invoke fails.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Configure the Lookup Tables

The recipe contains the following lookup tables. Edit the lookup tables as necessary.

- **ORACLE-BRT-SHOPIFY_S4HANA_SALES_ORDER_PROPERTIES**: This lookup table contains Sales Order properties for S/4HANA.

Property	Description	Value (Example)
SalesOrderType	Classification of sales orders.	OR
SalesOrganization	Specify the organizational unit responsible for sales activities.	1710
DistributionChannel	Specific distribution channel through which products are sold and distributed.	10
OrganizationDivision	Specific internal business division or category within the organization.	00
BusinessPartnerCategory	Specific category or classification of business partners.	22
Language	Select the language.	EN
BusinessPartnerIsBlocked	Flag indicating blocked business partners.	FALSE
Material	Products or items available for sale.	FG111
SalesOrderItemCategory	Categorization of individual items within an order.	TAN

- **ORACLE-BRT-SHOPIFY_METAFIELD_PROPERTIES**

Property	Description	Value
namespace	Specifies the category to which the property belongs.	order
key	Unique identifier for the property.	S4hanaOrderId
value_type	Specifies the data type of the key attribute.	integer

1. In the Lookups section, click the lookup name.
2. In the **Property** column, enter the name of the column.
3. In the **Value** column, enter the values as shown in the table below to which the property name must be mapped.

Field	Information to Enter
Hostname	Enter the Shopify host name. For example: <code>https://oicpm.myshopify.com</code>
Shopify REST API Version	Enter the OData Version. Example: 2023-01.

4. Click **Save**. If prompted, click **Save** again.
5. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must create an order in Shopify instance.

- a. Log in to Shopify instance.
- b. Navigate to Orders section.
- c. To create a new order, click **Create Order**.
- d. Search for the product in the **Product** field, then click **Browse**.
- e. Select the required product from the list, then click **Add**.
- f. In the customer field, select the **customer** to which the order has to be created.

Note

You can create a new customer if the customer you are looking for doesn't exist, which will then sync with SAP S/4 HANA.

- g. Select the Payment terms, then click **Create Order**.

Once the order is created, the integration **Oracle Shopify OData SAP S4Hana Sales Order Sync** gets triggered.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to SAP S/4 HANA and check for the sales orders created.
 - a. In the Search box, type Manage Sales Order, then click search icon.
 - b. In the Manage Sales Order window, click **Go**.

A list of orders created is displayed.

Related Documentation

- [Using the Shopify Adapter with Oracle Integration 3](#)
- [Using the OData Adapter with Oracle Integration 3](#)

Sync Process Automation Audit Data to Oracle Autonomous Transaction Processing

Use this recipe to synchronize Oracle Cloud Infrastructure Process Automation (Process Automation) audit data to Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe incrementally fetches and updates process audit, history and comments data from Process Automation to ATP database at scheduled intervals.

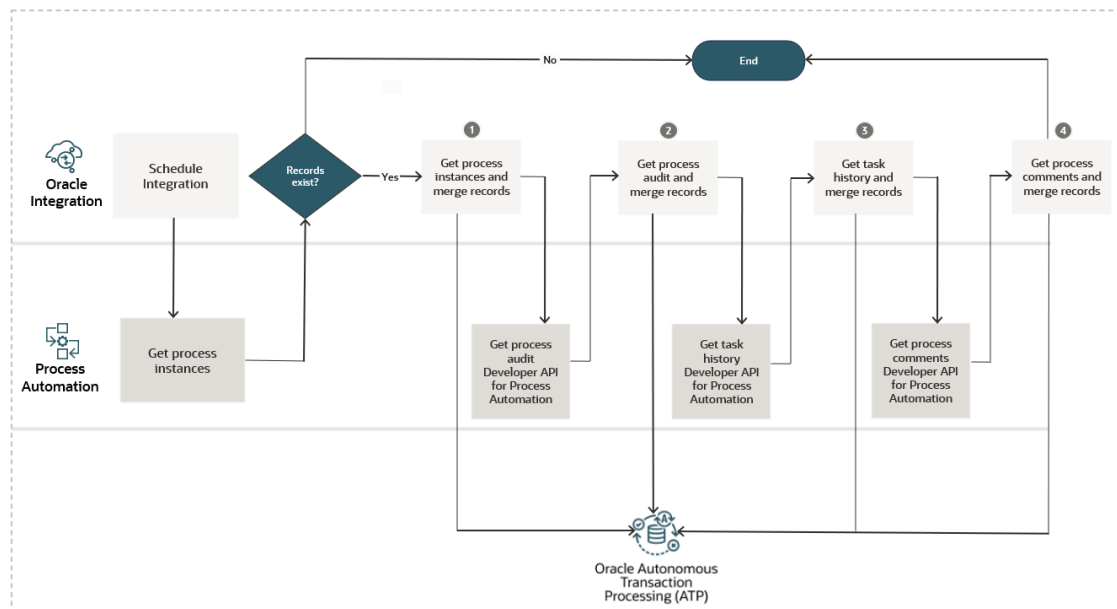
To use the recipe, you must install the recipe and configure the connections and other resources within it. When the scheduled integration flow of the recipe gets triggered, it fetches

process instance data from Process Automation and merges the records as needed. Then, the recipe uses Process Automation Developer APIs to get the process audit data, task histories and comments for those instances and merges the data. All processed and merged records are ultimately stored in Oracle Autonomous Transaction Processing.

System and Access Requirements

- Oracle Integration 3
- Process Automation
- An account on Process Automation with the Administrator role
- Oracle Autonomous Transaction Processing
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Recipe Schema



When the scheduled integration flow of the recipe is triggered, it fetches process instance details from Process Automation. If no process instance exists, the flow ends. Otherwise, the recipe continues to gather more details about the process instance. First, it uses Developer APIs for Process Automation to get process audit data from Process Automation, merges the audit data in Oracle Integration, and stores the audit data records in ATP database. Then, it uses Developer APIs for Process Automation to get task history details from Process Automation, merges the task history data in Oracle Integration, and stores the task history data records in ATP database. Finally, it uses Developer APIs for Process Automation to get process comments, merges the comments data in Oracle Integration, and stores the comments data records in ATP database. Thus, using this recipe you can extract, consolidate and store Process Automation instances' audit, task history and comments data in Oracle Autonomous Transaction Processing to boost analytic and operational efficiency.

Before You Install the Recipe

You must perform the following configuration tasks on your Process Automation and Oracle Autonomous Transaction Processing instances in order to successfully connect to these external systems using Oracle Integration and synchronize audit, history and comments data.

Configure Process Automation

To successfully connect to Process Automation using Oracle Integration and synchronize audit, history, and comment records, you must perform certain configuration tasks on the your Process Automation instance.

1. Configure an OAuth application for Process Automation.

Developer APIs for Process Automation and Process Automation REST endpoints are protected using OAuth token-based authentication.

- a. Sign in to the Oracle Identity Cloud Service administrator console.
- b. Click **Applications** and then **Add**.
- c. Select **Confidential Application** as the application type, enter the application name, and then click **Next**.
- d. Select **Configure this application as a client now**.
- e. Under **Allowed grant types**, select **Resource owner**.
- f. In **Client type**, select **Confidential**.
- g. In **Allowed scopes**, click **Add** and select your Process Automation instance from the list. Your instance name will be the instance ID. After the instance is selected, the application shows the configured scopes for the instance. You can select the required scopes (for example, /process).
- h. Click **Next** and skip the rest of the sections, and then click **Finish**.
A confidential OAuth application is created.
- i. Copy and save the **Client ID** and **Client Secret** for the application that is displayed and click **Close**.
- j. Click **Activate**, then click **OK** to confirm that you want to activate the application.
- k. Assign the **ServiceDeveloper** or the **ServiceAdministrator** role. For more information, see [Assign the application to the ServiceDeveloper role](#) in *Using Oracle Cloud Infrastructure Process Automation*.

2. Assign the **Process Application Administrator** global role to the application you configured in the previous step.

- a. Sign in to your Process Automation instance as an administrator.
- b. From the Process Automation Workspace navigation pane, click **Administration**, and then click **Roles**.
- c. Under the **Global** tab, select **Process Application Administrator**.
- d. In the **Search by** field, select **External applications** and then search and select the OAuth application that you configured.
- e. Click **Save**.

Configure Oracle Autonomous Transaction Processing

Sign in to your Oracle Autonomous Transaction Processing instance as an **Administrator** and perform the following tasks.

1. Perform general configuration tasks. See Prerequisites for Creating a Connection in *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.

2. In SQL developer, run the script to create schema.

The following script creates a schema in ATP database: [01_Create OPA Schema.sql](#)

3. In SQL developer, run the script to create database tables and necessary indexes.

The following script creates the database tables listed below:

[02_OPA_AuditSync_DBOjects.sql](#)

- XX_OPA_PROCESS_INSTANCE: Stores high-level process instance information.
- XX_OPA_PROCESS_ACTIVITY_INSTANCE: Stores process instance activity details.
- XX_OPA_PROCESS_TASK_HISTORY: Stores process task history details.
- XX_OPA_PROCESS_COMMENTS: Stores process comments details.
- XX_INTERFACE_LASTRUNTIME: Stores details about the last runtime when the data synchronization was performed.

4. In SQL developer, run scripts to create database package.

- [04_XX_OPA_AUDIT_SYNC_PKG_Specification.sql](#)

- [05_XX_OPA_AUDIT_SYNC_PKG_Body.sql](#)

On running the scripts, the database package XX_OPA_AUDIT_SYNC_PKG with four procedures is created.

- a. MERGE_OPA_PROCESS_INSTANCES
- b. MERGE_OPA_ACTIVITY_INSTANCE
- c. MERGE_OPA_TASK_HISTORY
- d. MERGE_OPA_PROCESS_COMMENTS

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Process Automation Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection type	Select REST API Base URL .
Connection URL	Enter your Process Automation instance URL.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select OAuth Resource Owner Password Credentials .
Access Token URI	Enter the access token URI.
Client Id	Enter the client Id that you obtained. See Configure Process Automation .
Client secret	Enter the client secret that you obtained.
Username	Enter the username of the account created for Oracle Integration on the Process Automation instance,
Password	Enter the password of the account created for Oracle Integration on the Process Automation instance,

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See [Configure Oracle Autonomous Transaction Processing](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Select the check box, then click Upload to upload the wallet file. See Configure Oracle Autonomous Transaction Processing .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **OPA_Audit_Sync_LKP** lookup table and enter appropriate values for the lookup keys.

1. In the Lookups section, click the lookup name.
2. Edit the lookup keys as per your requirements.

Key	Description
Email_Notification_To	Enter the email address of the recipient to whom error notifications have to be sent.
Email_Notification_From	Enter the email address of the sender who sends error notifications.
InterfaceName	Specify the name of the interface. For example: OPA_Audit_Sync.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Verify if Process Automation audit, task history and comments data is synchronized into the ATP database tables.

Related Documentation

- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*
- *Using the REST Adapter with Oracle Integration 3*
- [Developer API for Oracle Cloud Infrastructure Process Automation](#)

Sync Project Details from Oracle Primavera Cloud to PostgreSQL Using Amazon SQS

Use this recipe to synchronize project details from Oracle Primavera Cloud to PostgreSQL using Amazon Simple Queue Service (Amazon SQS).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is a scheduled orchestration recipe that includes two integrations to synchronize project details from Oracle Primavera Cloud to PostgreSQL using Amazon SQS.

To use the recipe, you must install the recipe and configure the connection and other resources within it. When triggered, the first integration flow of the recipe - **Oracle Primavera AWS SQS ProjectSync** extracts the project details from Oracle Primavera Cloud and subsequently publishes the details to Amazon SQS. This in turn triggers the second integration flow of the recipe - **AWS SQS PostgreSQL DB ProjectSync** which consumes the published project details from Amazon SQS as messages and updates the same to the PostgreSQL database.

System and Access Requirements

- Oracle Integration, Version 24.04 or higher
- Oracle Primavera Cloud
- An account on Oracle Primavera Cloud with the Administrator role
- An IAM user account on Amazon Web Services (AWS) with **AmazonSQSFullAccess** permission
- A PostgreSQL database
- An account on the PostgreSQL database with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Amazon SQS, Oracle Primavera Cloud and PostgreSQL instances to successfully connect to these external systems using Oracle Integration and sync project details.

Configure Oracle Primavera Cloud

To access Oracle Primavera Cloud from Oracle Integration, you must have a separate user account on Oracle Primavera Cloud.

Log in to your Oracle Primavera Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the username and password set for the account. You must use the credentials of this user account to connect to Oracle Primavera Cloud from Oracle Integration.
2. The user account must have **Read** permission for the project object.

Configure Amazon SQS

To access Amazon SQS using Oracle Integration, you must perform the following configuration tasks.

1. Create an AWS Account
2. Create an IAM User and Obtain the Access Key and Secret Access Key

In addition, review the connection properties and securities that you have to configure. See *Create a Connection in Using the Amazon Simple Queue Service (SQS) Adapter with Oracle Integration 3*.

Configure PostgreSQL

To access the PostgreSQL database from Oracle Integration and sync project details, you must perform certain configurations on your PostgreSQL instance.

For general configuration tasks, see *Prerequisites for Creating a Connection in Using the PostgreSQL Adapter with Oracle Integration 3*.

In addition, you have to create a database table. To create a database table:

1. Log in to your PostgreSQL Workbench instance as an **Administrator**.
2. On the left navigation pane, expand **Schemas** and then **public**.
3. Create the database table in either of the two ways given below.
 - a. Right-click on the public schema.
 - b. Click **Query Tool**.
 - c. Paste the query on the Query tool and then execute.or,
 - a. Under public schema, right-click **Tables**, select **Create** and then **Tables**.
 - b. In the Create-Table dialog, specify the name of the table as `primavera_projects_tab`.
 - c. Right-click on the empty table, and under **Create** select **Column**. Download [primavera_projects_tab.sql](#) for sample data to populate in the columns of the database table.
 - d. Click **Save**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Primavera Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle Primavera Cloud host name. For example: `https://your_domain_name.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Oracle Primavera Cloud Login .
Username	Enter the username of the account created for Oracle Integration on the Oracle Primavera Cloud instance. See Configure Oracle Primavera Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle Primavera Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Amazon SQS Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the AWS region name. For example: `ap-south-1`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Amazon Signature Version 4 .
Access Key	Enter the access key obtained from the AWS instance. See Configure Amazon SQS .
Secret Key	Enter the secret key obtained from the AWS instance.
AWS Region	Select the same AWS region that you entered in step 2 (<code>ap-south-1</code>).

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the PostgreSQL Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the PostgreSQL host name. See Configure PostgreSQL .
Port	Enter the PostgreSQL port.
Database Name	Enter the name of the database.

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of your PostgreSQL instance.
Password	Enter the password of your PostgreSQL instance.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the **ORACLE-BRT-AWS_SQS_PROPERTY** lookup table and enter appropriate values.

- In the Lookups section, click the lookup name.
- In the **Property_Name** column, enter the name of the column.
- In the **Property_Value** column, enter the value as shown in the table below.

Property	Description	Value
QueueURL	Specifies the Queue URL.	<code>https://sqs.<AWS Region>.amazonaws.com/<AWS Account ID>/<Queue Name>.fifo</code>

- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for both the scheduled integration flows (**Oracle Primavera AWS SQS ProjectSync** and **AWS SQS PostgreSQL DB ProjectSync**). You can update the integration properties only after activating the integration flows.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the following integration properties with appropriate values.
 - **ToEmail**: This integration property holds the email address to which error notifications are sent. Enter an email address of your choice.
 - **RetryCount**: This integration property holds the number of attempts made in case of errors. This could be due to network glitch or any other connectivity issues.
 - c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.
3. Run the **Oracle Primavera AWS SQS ProjectSync** integration flow to extract project details from Oracle Primavera Cloud.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the **Oracle Primavera AWS SQS ProjectSync** integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the project id (available projects in Oracle Primavera Cloud to be extracted) in the schedule parameter.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The recipe now extracts the project details from Oracle Primavera Cloud and publishes the same to Amazon SQS.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Run the **AWS SQS PostgreSQL DB ProjectSync** integration flow to consume the messages from Amazon SQS and update PostgreSQL.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the **AWS SQS PostgreSQL DB ProjectSync** integration flow, then select **Run**.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution. The recipe now consumes messages from Amazon SQS and updates PostgreSQL.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Log in to your PostgreSQL instance and check if the project details are populated in the database table.

- a. In the left navigation pane, right-click the **primavera_projects_tab** table.
- b. Select **View/Edit Data**.
- c. Click **All Rows**.

Related Documentation

- [Using the Oracle Primavera Cloud Adapter with Oracle Integration 3](#)
- [Using the PostgreSQL Adapter with Oracle Integration 3](#)

Sync Salesforce Accounts with Oracle CX Sales and Oracle Autonomous Transaction Processing

Use this recipe to synchronize accounts in Salesforce with Oracle CX Sales and Oracle Autonomous Transaction Processing.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe demonstrates how to use events to publish and subscribe account data from Salesforce to Oracle CX Sales and Oracle Autonomous Transaction Processing (ATP).

To publish and subscribe data using **Events**:

- A JSON-formatted or XML schema event is created.
- The **Publish event** action is used to publish the event created.
- The **Event** integration pattern is used to create integrations that subscribes to the published events. The subscribing integration event is triggered when the integration publishing the event is run. Multiple integrations can subscribe to the same published event.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When an account is created in Salesforce, the first integration flow **Subscribe Accounts from SFDC** gets triggered. The integration subscribes to the outbound message from Salesforce and publishes the account event using the publish event action. This in turn triggers the following integration flows: **CX Subscription for Accounts** and **ATP Subscription for Accounts**. These integrations are created using the event integration pattern that subscribes to the published event. Subsequently, the integration flows creates a corresponding account in Oracle CX Sales and ATP respectively.

If required, you can add more integrations that subscribe to the account create event. In this way, you can synchronize account data from a source application to multiple target applications.

System and Access Requirements

- Oracle Integration 3

- Salesforce
- Oracle CX Sales
- Oracle Autonomous Transaction Processing
- An account on Salesforce with the Administrator role
- An account on Oracle CX Sales with the Administrator role
- An account on Oracle Autonomous Transaction Processing with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Oracle Autonomous Transaction Processing instances to connect to these external systems using Oracle Integration and sync accounts.

Configure Salesforce

To access Salesforce from Oracle Integration and retrieve information about Salesforce accounts, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce account with active as **Yes** is created.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:

- i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
 - b. Click **New User** in the resulting page.
 - c. On the New User page:
 - i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.

The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.
 - d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
 - e. Set a password and note the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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. See Find Your Current Salesforce API Version.
4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization.
5. Create an outbound message. The outbound message triggers the integration flow of this recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.
 - c. On the New Outbound Message page:
 - i. In the **Object** field, select **Account** and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the message.
 - In the **Endpoint URL** field, enter the following URL: `https://<OIC_FQDN:OIC_SSL_PORT>/ic/ws/integration/v2/flows/salesforce/project/ORCL-R-PUB-SUB-SOURCE-TARGET/SUBSCRIB_ACCOUNTS_FROM_SFDC/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 **Account 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 account is created.

- 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 **Account** in the **Object** field, and click **Next**.

- ii. Enter the following details for the new workflow rule:

- Enter the rule's name.
- Optionally, enter a description for the rule.
- In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
- In the Rule Criteria section, select the **formula evaluates to true** option, and define the following rule criteria:

Account: Active EQUALS Yes

- Click **Save & Next**.

- iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.

- iv. On the Select Existing Actions page:

- Select **Outbound Message** in the **Search** field.
- In the **Available Actions** box, select the outbound message you created previously, for example,

Outbound Message: <name of your outbound message>

.

- Click **Add** to move your selection to the **Selected Actions** box.
- Click **Save**.

- v. Click **Done**.

The new workflow rule is created and displayed on your browser.

- d. Click **Activate** to activate the rule.

Configure Oracle Autonomous Transaction Processing

To successfully connect to Oracle Autonomous Transaction Processing using Oracle Integration, and synchronize account data from Salesforce to ATP you must perform certain configurations on your Oracle Autonomous Transaction Processing instance.

1. Perform the general configuration tasks for creating a connection. See *Prerequisites for Creating a Connection in Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.
2. Run the following script to create the necessary objects:

```
CREATE TABLE "ADMIN"."XX_ACCOUNTS"
(
  "ACCOUNT_ID" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "NAME" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
  "TOTAL_EMPLOYEES" NUMBER(*,0),
  "CONTACT_NUMBER" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
  "SOURCE" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP"
) DEFAULT COLLATION "USING_NLS_COMP" ;

CREATE UNIQUE INDEX "ADMIN"."XX_ACCOUNTS_PK" ON "ADMIN"."XX_ACCOUNTS"
("ACCOUNT_ID")
;

ALTER TABLE "ADMIN"."XX_ACCOUNTS" ADD CONSTRAINT "XX_ACCOUNTS_PK" PRIMARY
KEY ("ACCOUNT_ID")
USING INDEX ENABLE;
```

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. See Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.

Field	Information to Enter
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.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle ATP Connection

- In the Connections section, click the connection name.
- In the Properties section, specify the database service name in the **Service Name** field. See [Configure Oracle Autonomous Transaction Processing](#).
- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Select the check box, then click Upload to upload the wallet file. See Configure Oracle Autonomous Transaction Processing .
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle CX Sales Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on the Oracle CX Sales instance.
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, create an account on your Salesforce instance.

- a. Log in to your Salesforce instance.
- b. On the Salesforce Setup page (in the Classic UI), click the **Accounts** tab and choose **+ New Account**.
- c. On the New Account dialog box, enter the relevant details such as Account Name, Phone, Account Number, Employees and so on, and then click **Save**.

You've successfully triggered the integration flows of the recipe. The recipe now creates corresponding accounts in Oracle CX Sales and ATP for the new account created on Salesforce.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle CX Sales instance and check if a new account corresponding to the new Salesforce account is created.
5. Log in to your ATP instance and check the `XX_ACCOUNTS` table for the new account.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Sync Salesforce Accounts with ServiceNow Companies

Use this recipe to synchronize Salesforce accounts with ServiceNow companies.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a company in ServiceNow when a new account is created in Salesforce. Also, whenever an account in Salesforce is updated, the recipe updates the corresponding company in ServiceNow.

To use this recipe, you must install the recipe and configure the connections and other resources within the recipe.

When a Salesforce account is created or updated, the integration flow of the recipe is triggered, and it creates or updates the corresponding company in ServiceNow. Basic data such as account name, phone, employees, and billing address are synchronized between Salesforce and ServiceNow platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- ServiceNow
- An account on ServiceNow with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and ServiceNow instances in order to successfully connect to these external systems using Oracle Integration and synchronize Salesforce accounts with ServiceNow companies.

Configure Salesforce

To access Salesforce from Oracle Integration and retrieve data about Salesforce accounts that are created or updated, you must perform certain configurations on your Salesforce instance.

Create a user account on Salesforce for Oracle Integration. You'll use the credentials of this user account while configuring the Salesforce connection in Oracle Integration.

Also, create a custom field for Salesforce account records to store ServiceNow company IDs, and configure the generation of change-event notifications for the Accounts entity.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. See [Toggle or switch between Lightning Experience and Salesforce Classic](#).

Log in to your Salesforce instance as an Administrator and perform the following tasks:

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. In the **Existing Profile** field, select **Standard User**.
 - ii. Enter a name for the new profile (for example, API Enabled), and click **Save**.
The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the **Administrative Permissions** section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the **Standard Object Permissions** section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked.
 - Select the **ViewAll** check box.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.

The email address you enter is automatically populated in the **Username** field. Note this username.

- iii. In the **User License** field, select **Salesforce**.
- iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
- v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.

- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note it.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.
 - ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the **Quick Links** section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.

A new security token is sent to the email address associated with the account. Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
- 3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
- 4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
- 5. Create a custom field for contact records.

This recipe uses unique IDs associated with company records in ServiceNow to synchronize ServiceNow companies with Salesforce accounts. Create a custom field for Salesforce account records to hold the ServiceNow IDs.

- a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Customize**, then **Accounts**, and then click **Fields**.

- b. On the Account Fields page, scroll to the **Account Custom Fields & Relationships** section and click **New**.
6. Configure the generation of change-event notifications for the Account entity.
With this configuration, when a new account is created or updated in Salesforce, a change-event notification is generated to trigger the recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the **Build** section in the left navigation pane.
 - ii. Expand **Develop**, then click **Change Data Capture**.
 - b. On the Change Data Capture page, from the list of available entities, select **Accounts**, and then click the right arrow.
 - c. Select the account, click **Save**.

Configure ServiceNow

You can use admin account credentials or custom account credentials while configuring the ServiceNow connection in Oracle Integration.

If you do not want to use an admin account, create a custom integration user and assign the required role and permissions to the user. For information about creating a custom user, see *Prerequisites for Creating a Connection in Using the ServiceNow Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .

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** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Configure the Oracle ServiceNow Connection

- In the Connections section, click the connection name.
- In the Connection Properties section, enter the ServiceNow host name. For example:
`https://instance_name.servicenow.com`.
- In the Security section, enter the following details:

Field	Information to Enter
Username	Enter your ServiceNow username.
Password	Enter your ServiceNow password.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
- To return to the project workspace, click **Go back** .

Update Integration Property

Update the **EmailTo** integration property with an appropriate value.

- In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
- In the Update property values panel, update the integration properties with appropriate values.
 - EmailTo:** This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
- Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must create an account in Salesforce.

 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page, click the **Accounts** tab.
 - c. On the Accounts Home page, click **New**.
 - d. On the New Account page, enter the account details, such as name, description, and so on. Click **Save**.

You've now created an account record in Salesforce and triggered the recipe.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your ServiceNow instance and check if a new company has been created.
 - a. In the left navigation pane, under **Organization**, click **Companies**.
 - b. On the Companies page, search for the newly created company.
 - c. Click the company record's name to view details. Check if the newly created company's data matches with the Salesforce account that you created earlier.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the ServiceNow Adapter with Oracle Integration 3*

Sync Salesforce Accounts with Shopify Customers

Use this recipe to synchronize Salesforce accounts with customer records in Shopify.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is a bidirectional, application-driven recipe. It creates or updates a corresponding customer record in Shopify when an account record is created or updated in Salesforce. Similarly, it creates or updates a corresponding account record in Salesforce when a customer record is created or updated in Shopify.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe.

When an account is created or updated in Salesforce, the **Oracle Salesforce Shopify Account To Customer Sync** integration flow of the recipe is triggered, and it creates or updates a corresponding customer record in Shopify. Similarly, when a customer record is created in Shopify, the **Oracle Shopify Salesforce Customer To Acc Create** integration flow of the recipe is triggered, and it creates a corresponding account in Salesforce. Further, when Shopify customer records are updated, the **Oracle Shopify Salesforce Customer To Acc Update** integration flow is triggered, which propagates the updates to corresponding

Salesforce account records. Basic data, such as name, email, note, phone, and address, are synchronized between the Salesforce and Shopify platforms.

System and Access Requirements

- Oracle Integration, Version 22.2.1 or higher
- Salesforce, Version 48.0 or later
- An account on Salesforce with the Administrator role
- Shopify
- An account on Shopify with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and Shopify instances in order to successfully connect to these external systems using Oracle Integration and synchronize records between them.

Configure Salesforce

To access Salesforce from Oracle Integration and create or read account records, you must perform certain configurations 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.

Also, identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the **Oracle Salesforce Shopify Account To Customer Sync** integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce account is created. Finally, you must create a custom field for Salesforce account records to store Shopify customer IDs.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
 - a. On the Salesforce Setup page:
 - i. Expand **Manage Users** under the Administer section in the left navigation pane.
 - ii. Click **Profiles**.
 - b. On the Profiles page, click **New Profile**.
 - c. On the resulting page:
 - i. Select **Standard User** in the **Existing Profile** field.
 - ii. Enter a name for the new profile, for example, **API Enabled**, and click **Save**.

The new profile is now saved, and the Profile Detail page of the new profile is displayed.

- d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the Accounts row, leave the **Read** box checked. Additionally, select the **Create, Edit, and Delete** boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the API Enabled profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user, for example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.
The email address you enter is automatically populated in the **Username** field. Note this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the user name at the top of the page to open a menu.

- ii. Click **My Settings** in the menu.
 - iii. On the My Settings page, in the Quick Links section, click **Edit my personal information**.
 - iv. On the resulting page, click **Reset My Security Token** in the left navigation pane.
 - v. Click the **Reset Security Token** button.
A new security token is sent to the email address associated with the account.
Note the security token.
 - vi. On the Salesforce instance, click the user name again and select **Logout** from the menu. Log back in as the **Administrator**.
 3. Identify your current Salesforce API version. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
 4. Identify your Salesforce instance type. See Identify the Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
 5. On your Salesforce instance, create an outbound message to trigger the **Oracle Salesforce Shopify Account To Customer Sync** integration flow of this recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.
If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.
 - c. On the New Outbound Message page:
 - i. Select **Account** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the message.
 - In the **Endpoint URL** field, enter the following URL:

`https://<OIC_FQDN:OIC_SSL_PORT>/ic/ws/integration/v2/flows/salesforce/project/ORCL-R-SFDC_SHOPIFY_ACC_CUSTOMER/ORCL-R-SFDC_ACC_SYN_TO_SHOPY_CUS/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).
- 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 account is created or updated.
 - a. On the Salesforce Setup page:

- i. Scroll to the Build section in the left navigation pane.
- ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.

- b. Click **New Rule** in the resulting page.
- c. On the New Workflow Rule page:
 - i. Select **Account** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:
 - Enter the rule's name.
 - Optionally, enter a description for the rule.
 - In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
 - In the Rule Criteria section, select the **formula evaluates to true** option, and define the following rule criteria:

(Account: Account Name STARTS WITH Shop) AND (Account: ShopifyCustomer_id EQUALS empty)
 - Click **Save & Next**.
 - iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
 - iv. On the Select Existing Actions page:
 - Select **Outbound Message** in the **Search** field.
 - In the **Available Actions** box, select the outbound message you created previously, for example, Outbound Message: <name of your outbound message>.
 - Click **Add** to move your selection to the **Selected Actions** box.
 - Click **Save**.
 - v. Click **Done**.

The new workflow rule is created and displayed on your browser.

- d. Click **Activate** to activate the rule.
7. Create a custom field for account records.

This recipe uses unique IDs associated with customer records in Shopify to synchronize Shopify customers with Salesforce accounts. Create a custom field for Salesforce account records to hold the Shopify IDs.

- a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Accounts**, and then click **Fields**.
- b. On the Account Fields page, scroll to the Account Custom Fields & Relationships section and click **New**.
- c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.

- ii. Enter the following details for the new custom field:
 - Enter `ShopifyCustomer_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**.
- d. Repeat steps a, b, and c above to create another custom field called **Email**. But exclude the sub-step to select the **External ID** check box, under step c.

On the Account Fields page, you can see the new field added under the Account Custom Fields & Relationships section.

Configure Shopify

To configure Shopify, see Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production , Sandbox , or Government based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .

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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure Shopify

To configure Shopify, see Prerequisites for Creating a Connection in *Using the Shopify Adapter with Oracle Integration 3*.

Update Integration Property

Update the **EmailTo** integration property with appropriate value in the recipe's integration flows.

1. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the following integration property with appropriate value.
 - **EmailTo:** This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.
3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe to create or update Shopify customers for new or updated Salesforce account records.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page (in the Classic UI), click the **Accounts** tab.
 - c. Create a new account.
 - i. On the Accounts Home page, click **New**.

- ii. On the New Account page, enter the account details, such as name, description, and so on, and ensure that the **ShopifyCustomer_id** field is blank. Click **Save**.
 - d. Or, update an account.
 - i. On the Accounts Home page, select **All Accounts** in the **View** field, and click **Go!**. All the accounts in your Salesforce instance are displayed.
 - ii. Click an account record to open it.
 - iii. On the account's page, in the Account Detail section, click **Edit**. Further, scroll to the Notes & Attachments section, click **New Note**, enter a title and body message for the note, and click **Save**.
 - You've now successfully triggered the recipe, which creates or updates a corresponding customer record in Shopify.
 3. Run the recipe to create or update accounts in Salesforce for Shopify customer records.
 - a. Log in to the Shopify instance.
 - b. On the left navigation pane, click **Customers**.
 - c. Create a new customer record.
 - i. In the Customers window, click **Add Customer**.
 - ii. Enter the following details.
 - First Name, Last Name, Email, Phone, and Address.
 - Select the **Collect Tax** check box if the customer is not tax exempted.
 - Add additional notes if necessary.
 - d. Or, update an existing customer record.
 - i. In the Customers window, select a customer record from the list of available records, and click **Edit**.
 - ii. Update the any detail of the customer record, and click **Save**.
 - You have now successfully triggered the recipe, which creates or updates a corresponding account record in Salesforce.
 4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
 5. Log in to your Shopify instance and check for the new or updated customer records.
 - a. On the home page, select your Shopify store, for example, `orcl-shop-connector-dev-store`.
 - b. On the store page, click **Customers** on the left navigation pane. The list of the customers is displayed, which contains the corresponding records for the accounts created or updated in Salesforce.
 - c. To view a customer record's details, click its name.
 6. Log in to your Salesforce instance and check for the new or updated account records.
 - a. On the Salesforce Setup page (in the Classic UI), click the **Accounts** tab.
 - b. On the Accounts Home page, select **All Accounts** in the **View** field. Click **Go!** if necessary.

The list of accounts is displayed, which contains the corresponding records for the customers created or updated in Shopify.
 - c. To view an account record, click its name.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the Shopify Adapter with Oracle Integration 3*

Sync Salesforce Accounts with Workday Customers

Use this recipe to synchronize Salesforce accounts with Workday customers.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates or updates a customer in Workday whenever an account is created or updated in Salesforce. In addition, it creates a Workday invoice for the closed Salesforce opportunity and posts a Slack notification if the invoice is in unpaid status.

To use the recipe, you must install the recipe project and configure the connections and other resources within it. When an account is created or updated in Salesforce, the application-driven integration flow **Oracle Salesforce Workday AccountToCustomer Sync** is triggered and it creates or updates a corresponding customer in Workday. Subsequently, whenever a Salesforce opportunity is closed, the application-driven integration flow **Oracle Salesforce Workday OpportunityToInvoice Syn** is triggered and it creates an invoice for the corresponding account in Workday. Further, you can run the scheduled integration flow **Oracle Workday Slack SendMessage** of the recipe manually or specify an execution schedule for it. When triggered, the **Oracle Workday Slack SendMessage** integration flow posts a message in Slack if the status of the invoice is unpaid.

In the **Oracle Salesforce Workday AccountToCustomer Sync** integration flow, basic data such as account name, account Id, account type, billing address, billing city, billing state/province, billing zip/postal code, and billing country are synchronized between the Salesforce and Workday platforms. Similarly, in the **Oracle Salesforce Workday OpportunityToInvoice Syn** integration flow, basic data such as customer reference, opportunity Id (customer PO number in Workday), company reference, sales item reference and so on are mapped between the Salesforce and Workday platforms.

System and Access Requirements

- Oracle Integration, Version 24.08 or higher
- Salesforce
- An account on Salesforce with the Administrator role
- Workday
- An account on Workday with the Administrator role
- Slack
- An account on Slack with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce, Workday and Slack instances to successfully connect to these external systems using Oracle Integration, and synchronize Salesforce accounts with Workday customers.

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve information about the Salesforce accounts, 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, configure change data capture to generate change event notifications on the default standard channel.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI.

1. Create an API-enabled custom role. You'll assign this role to the user account you'll subsequently create for Oracle Integration.
 - a. On the Salesforce Setup page:
 - i. Expand **Manage Users** under the Administer section in the left navigation pane.
 - ii. Click **Profiles**.
 - b. On the Profiles page, click **New Profile**.
 - c. On the resulting page:
 - i. Select **Standard User** in the **Existing Profile** field.
 - ii. Enter a name for the new profile, for example, **API Enabled**, and click **Save**. The new profile is now saved, and the Profile Detail page of the new profile is displayed.
 - d. Click **Edit** on the Profile Detail page.
 - e. On the Profile Edit page:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following action.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - 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.
 - iii. The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iv. In the **User License** field, select **Salesforce**.
 - v. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - vi. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.
 - vii. 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 into the Salesforce instance with the new account.

Note

If you're shown the Lightning Experience UI, switch to the Salesforce Classic UI.

- f. Generate a security token for the new user account. You'll need this security token along with the password to access Salesforce using Oracle Integration.
 - i. Stay signed in as the new user and click the username 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.
 - vi. A new security token is sent to the email address associated with the account. Note down the security token.
 - vii. On the Salesforce instance, click the username 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:

```
CopySalesforce.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, select an entity (**Accounts** in this case) to which change event notifications has to be generated. To select the entity,
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Develop**, then click **Change Data Capture**.
 - b. On the Change Data Capture setup page, from the list of available entities, click **Accounts**, then click the right arrow icon.
 - c. After selecting the entity, click **Save**.
6. Create a custom field for Salesforce to store `Workday_Customer_ID`.

This recipe uses unique IDs associated with Workday and Salesforce instances to synchronize accounts/customers.

- a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Accounts**, and then click **Fields**.
 - b. On the Account Fields page, scroll to the Account Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:
 - i. Find and select the **Text** radio button and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter `Workday_Customer_ID` as the field label. Note that the field name is automatically populated based on the label you enter.
 - Specify the character length.
 - Optionally, enter a description for the new field.
 - Select the checkbox next to **Set this field as the unique record identifier from an external system**.
 - Click Next.

- d. 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**.
 - e. Leave all the layout check boxes selected and click **Save**.
On the Account Fields page, you can see the new field added under the Account Custom Fields & Relationships section.
7. Create custom fields under Salesforce opportunity object to store the **WorkdayInvoiceId**. To create a custom field for Salesforce opportunities.
- a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Opportunities**, and then click **Fields**.
 - b. On the Opportunity Fields page, scroll to the Opportunity Custom Fields & Relationships section and click **New**.
 - c. On the New Custom Field page:
 - i. Enter the following details for the new custom field:
 - Enter `Workday_Invoice_Id` as the field label. Note that the field name is automatically populated based on the label you enter.
 - Specify the character length.
 - Optionally, enter a description for the new field.
 - Click the checkbox next to **Set this field as the unique record identifier from an external system**.
 - Click **Next**.
 - 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**.
 - Leave all the layout check boxes selected and click **Save**.
On the Opportunity Fields page, you can see the new field added under the Opportunity Custom Fields & Relationships section.
8. Create custom fields for the platform events and update the apex trigger code in Salesforce for the integration **Oracle Salesforce Workday OpportunityToInvoice Syn**. To create the custom fields for platform events:
- a. On the Salesforce setup page, in the left navigation pane, navigate to **Build**, then **Develop**, then **Lightning Components**, and then to **Platform Events**.
 - b. In the Platform Events page, click **New Platform Event**.
 - c. In the New Platform Event page, complete the standard fields, and optionally add a description.
 - d. For Publish Behavior, choose when the event message is published in a transaction.
 - **Publish After Commit** to have the event message published only after a transaction commits successfully. Select this option if subscribers rely on data that the publishing transaction commits. For example, a process publishes an event message and creates a task record. A second process that is subscribed to the event is fired and expects to find the task record. Another reason for choosing this behavior is when you don't want the event message to be published if the transaction fails.

- **Publish Immediately** to have the event message published when the publish call executes. Select this option if you want the event message to be published regardless of whether the transaction succeeds. Also choose this option if the publisher and subscribers are independent, and subscribers don't rely on data committed by the publisher. For example, the immediate publishing behaviour is suitable for an event used for logging purposes. With this option, a subscriber might receive the event message before data is committed by the publisher transaction.
- e. Click **Save**.
- f. To add a field, in the Custom Fields & Relationships related list, click **New**.
- g. In the New Custom Field page, select the required **Data Type**, then click **Next**.
- h. Enter the following details for the new custom field:
 - i. Enter the field label. Note that the field name is auto populated based on the label you enter.
 - ii. Specify the character length.
 - iii. Optionally, enter a description for the new field.
 - iv. Click the checkbox next to **Always require a value in this field in order to save a record**.
 - v. Enter the additional details required for the respective fields, then click **Save**. You can see the new field added under the Custom Fields & Relationships section.
- i. Update the custom fields created for platform events in the apex trigger code.
Download and view the code: [ApexCode](#).

Configure Workday

To configure Workday, see Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.

Configure Slack

To configure Slack, see Prerequisites for Creating a Connection in *Using the Slack Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. See Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .
Password	Enter the password of the account created for Oracle Integration on Salesforce.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Workday Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Hostname	Enter the Workday host name. For example: <code>https://wd2-impl-services1.workday.com</code> .
Tenant Name	Enter the tenant name of the account created for Oracle Integration on Workday. See Configure Workday .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Workday Username Token Policy .
Workday Integration User	Enter the user name of the account created for Oracle Integration on Workday. See Configure Workday .
Password	Enter the password of the account created for Oracle Integration on Workday.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Slack Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Client ID	Enter the client ID obtained while configuring Slack. See Configure Slack .
Client Secret	Enter the client secret obtained while configuring Slack.
Scope	Enter the permission scopes you configured while creating and configuring a Slack app.

3. Click **Provide Consent**.
A new browser window opens to approve access to Slack.
4. Click **Allow**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
7. To return to the project workspace, click **Go back** .

Configure the Lookup Tables

Edit the lookup tables of the recipe and enter appropriate values.

1. In the Lookups section, click the lookup name.
2. The recipe contains the following two lookup tables. In the lookup tables, enter appropriate values to which the property names must be mapped.
 - a. Edit the **ORACLE-BRT-SFDC_WD_Lookup** table.

SFDC	WD
Customer - Direct	CUSTOMER_CATEGORY-3
USA	USA
CAN	CAN
New York	USA-NY
Canada	CAN-BC

- b. Edit the **ORACLE-BRT-SFDC_WORKDAY_COMPANY_LOOKUP** table.

Property	WD	Description
CompanyReferenceID	GMS_USA_company	Represents the unique identifier for a company reference.
WorktagsID	10000	Represents the unique identifier for worktags used to categorize and track transactions.

Property	WD	Description
PaymentTermsID	NET_30	Represents the terms of payment.
PaymentTypeID	EFT	Represents the method of payment.
TxCodeID	TAX_CODE-2	Represents the tax code identifier used for taxation purposes.
TenantBillingID	COMMUNICATION_USAGE_BEHAVIOR_TENANTED-16-10	Represents the billing identifier for tenant communications usage behavior.
TenantShippingID	COMMUNICATION_USAGE_BEHAVIOR_TENANTED-16-11	Represents the shipping identifier for tenant communications usage behavior.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Update Integration Properties

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update property values panel, update the integration properties with appropriate values.
 - a. **User:** This integration property holds the user Id of the Salesforce account user to avoid recurring updates.
This property has to be updated in the **Oracle Salesforce Workday AccountToCustomer Sync** integration flow.
 - b. **Email:** This integration property holds the email id to which error notifications have to be sent.
This property has to be updated in the **Oracle Salesforce Workday AccountToCustomer Sync** and **Oracle Salesforce Workday OpportunityToInvoice Syn** integration flows.
3. Click **Submit**.
A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update the integration property for the **Oracle Workday Slack SendMessage** integration flow.
 - a. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the following integration property with appropriate value.

- **ChannelName:** This integration property holds the name of the Slack channel to which invoice status notifications must be sent.
- c. Click **Submit**.
A message confirms that the integration properties have been updated successfully.
- 3. Run the recipe.
 - a. Run the **Oracle Salesforce Workday AccountToCustomer Sync** integration flow.
To run the integration flow, you have to create or update an account in your Salesforce instance.
Create a new Salesforce account:
 - i. Log in to your Salesforce instance.
 - ii. On the Salesforce Setup page, click the **Accounts** tab.
 - iii. To create a new account, click **New**.
 - iv. In the New Account window, enter the account details such as account name, billing address, billing city, billing state/province, billing zip/postal Code, billing country, and any additional data if required. Click **Save**.

You've successfully created an account in Salesforce and triggered the **Oracle Salesforce Workday AccountToCustomer Sync** integration flow. The integration now creates a corresponding customer in Workday for the newly created Salesforce account.
 - b. Run the **Oracle Salesforce Workday OpportunityToInvoice Syn** integration flow.
To run the integration flow, you have to close a Salesforce opportunity.
 - i. Log in to your Salesforce instance.
 - ii. On the Salesforce Setup page, click the **Opportunities** tab.
 - iii. On the Opportunities Home page, select **All Opportunities** in the **View** field and click **Go!**.
 - iv. Find a sample opportunity record that hasn't been closed and click its name to open it.
 - v. In the Opportunity Detail section at the top, click **Edit**, then select **Closed Won** in the **Stage** field, and click **Save**.

You've successfully closed a Salesforce opportunity and triggered the **Oracle Salesforce Workday OpportunityToInvoice Syn** integration flow. The integration now creates an invoice in Workday for the corresponding Salesforce account.
 - c. Run the **Oracle Workday Slack SendMessage** integration flow to post a Slack notification if the status of the Workday invoice is unpaid.
 - i. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.
The Configure and run page is displayed, where you can specify a value for the **InvoiceDate** parameter. This parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the invoices with unpaid status after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm.
 - ii. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
5. Log in to your Workday instance and check for the new customer and the invoice created.
 - a. To view the customer created:
 - i. In the search field, type View Customer, then press **Enter**. A pop-up with the search field against customer appears.
 - ii. In the search field enter the **Workday_Customer_ID** copied from the Salesforce Accounts page, then click **OK**.
The customer details are displayed.
 - b. To view the invoice created:
 - i. In the search field, type the **Invoice Id** starting with wid: (Example: wid:invoicid123).
 - ii. Press **Enter**.
The invoice details appear.

Related Documentation

- [Using the Salesforce Adapter with Oracle Integration 3](#)
- [Using the Workday Adapter with Oracle Integration 3](#)
- [Using the Slack Adapter with Oracle Integration 3](#)

Sync Salesforce Cases with ServiceNow Incidents

Use this recipe to synchronize Salesforce cases with ServiceNow incidents.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates an incident in ServiceNow when a new case is created in Salesforce. Also, whenever a case in Salesforce is updated, the recipe updates the corresponding incident in ServiceNow.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a Salesforce case is created or updated, the integration flow of the recipe is triggered, and it creates or updates the corresponding incident in ServiceNow. Basic data details is synchronized between Salesforce cases and ServiceNow incidents.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Salesforce
- ServiceNow
- An account on Salesforce with the Administrator role
- An account on ServiceNow with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and ServiceNow instances in order to successfully connect to these external systems using Oracle Integration and synchronize Salesforce cases with ServiceNow incidents.

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve data about Salesforce contacts that are created or updated, you must perform certain configurations 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.

Log in to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.

- ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - In the **Price Books** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - In the **Products** row, leave the **Read** box checked. Additionally, select the **Create**, **Edit**, and **Delete** check boxes.
 - iii. Scroll to the end of the page and click **Save**.
2. Create a user account for Oracle Integration and assign the custom role created previously to this account.

Note

If you have already created a user account for Oracle Integration, you can assign the API-enabled custom role to the existing account.

- a. On the Profile Detail page of the **API Enabled** profile, click **View Users**.
- b. Click **New User** in the resulting page.
- c. On the New User page:
 - i. Enter a first name and last name for the user. For example, Integration User05.
 - ii. In the **Email** field, enter a valid email address.

The email address you enter is automatically populated in the **Username** field. Note down this username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.
- e. Set a password and note the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
 4. Identify your Salesforce instance type. See Identify Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
 5. Create an outbound message. The outbound message triggers the integration flow of the recipe.
 - a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.
 - b. Click **New Outbound Message** in the resulting page.
 - c. On the New Outbound Message page:
 - i. In the **Object** field, select **Case** and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the message.
 - In the **Endpoint URL** field, enter the following URL: `https://<OIC_FQDN:OIC_SSL_PORT>/ic/ws/integration/v1/flows/salesforce/ORCL-R-SFDC_CON_TO_SNOW_USER_SYC/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`).
 - Under **Case Object**, select all the required fields and click **Add**.
 - Click **Save**.
6. Create a workflow rule to send the outbound message (created previously) when a Salesforce contact is created or updated.

- a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.
If you're shown the Understanding Workflow page, click **Continue**.
- b. Click **New Outbound Message** in the resulting page.
- c. On the New Workflow Rule page:
 - i. Select **Case** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:
 - Enter the rule's name.
 - Optionally, enter a description for the rule.
 - In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
 - In the Rule Criteria section, define the following rule criteria:
Case: Number not equals null
 - Click **Save & Next**.
 - iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
 - iv. On the Select Existing Actions page:
 - Select **Outbound Message** in the **Search** field.
 - In the **Available Actions** box, select the outbound message you created previously, for example, Outbound Message: <name of your outbound message>.
 - Click **Add** to move your selection to the **Selected Actions** box.
 - Click **Save**.
 - v. Click **Done**.
The new workflow rule is created and displayed on your browser.
- d. Click **Activate** to activate the rule.

Configure ServiceNow

To access ServiceNow using Oracle Integration and create corresponding ServiceNow incidents for Salesforce cases that are created or updated, you must perform certain configurations on your ServiceNow instance.

Log in to your ServiceNow instance as an **Administrator** and execute the following tasks.

1. Create a custom user (for example, Integration user) and assign a custom role that has access to ServiceNow tables. See **Satisfy User and Role Requirements** under Prerequisites for Creating a Connection in *Using the ServiceNow Adapter with Oracle Integration 3*.
2. Enable web services and assign permissions to required tables.
 - a. On the ServiceNow home page, search for tables in the search box in the left pane, and click the **Tables** link under **System Definition** in the search results.

- b. Search for each of the following ServiceNow tables using the **Search** box or locate a table using the show/hide filter.
 - sys_soap_message
 - sys_soap_message_function
 - sys_script
- c. Click the table name in the search results.
- d. Locate and click the **Application Access** tab.
- e. Select the respective permission (refer to the following table for required permissions), and select the **Allow access to this table via web services** check box if it is not selected already.

Table Name	Permission
sys_soap_message	Read, Create, Update, Delete
sys_soap_message_function	Read, Create, Update, Delete
sys_script	Read, Create, Update, Delete

3. Enable multiple insert plugin to support insertMultiple in SOAP.
 - a. On the ServiceNow home page, search for System Definition.
 - b. Select **Plugins**.
 - c. Locate and click **Insert Multiple Web Service**.
 - d. If the status is inactive, click **Activate/Repair**.
4. Get the ServiceNow Caller ID.
 - a. On the ServiceNow home page, search for **User Administration** and then click **Users**.
 - b. On the Users page, select the custom user that you configured earlier in step 1.
 - c. Locate the value of the sys_id in the URL that appears when you select the user.
 - d. Note the sys_id value as you would need to provide it while configuring lookup table in Oracle Integration.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.

Field	Information to Enter
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .
3. In the Security section, enter the following details:
- | Field | Information to Enter |
|-----------------|--|
| Security Policy | Leave Salesforce Username Password Policy selected. |
| Username | Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce . |
| Password | Enter the password of the account created for Oracle Integration on Salesforce.
Note: To the password, you must also append the security token generated for the same account. |
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
- A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle ServiceNow Connection

- In the Connections section, click the connection name.
 - In the Connection Properties section, enter the ServiceNow host name. For example: `https://instance_name.servicenow.com`.
 - In the Security section, enter the following details:
- | Field | Information to Enter |
|----------|--|
| Username | Enter your ServiceNow username. See Configure ServiceNow . |
| Password | Enter your ServiceNow password. |
- Click **Save**. If prompted, click **Save** again.
 - Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
- A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and enter the ServiceNow Caller ID of your ServiceNow instance. After updating the ServiceNow Caller ID, the Salesforce values will be automatically mapped with the ServiceNow values.

- In the Lookups section, click the lookup name.
- Enter the ServiceNow caller ID of your ServiceNow instance in the **ServiceNow_Value** column, against the **ServiceNowCallerID** field. Note that the ServiceNow Caller ID is the

sys_id value that you obtained in step 4 while configuring the ServiceNow instance. See [Configure ServiceNow](#).

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.

To run the recipe, you must create a new case in Salesforce.

- a. Log in to your Salesforce instance.
- b. On the Salesforce Setup page, click the **Cases** tab.
- c. On the Cases Home page, click **New**.
- d. On the new case page, enter the required details and click **Save**.

You've now successfully triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your ServiceNow instance and check for the new incident created.
 - a. Navigate to the Incidents page.
 - b. To find the newly created ServiceNow incident, filter with the case number of the new Salesforce case. Enter the Salesforce case number in the field under the **Number** column.

The new ServiceNow incident is listed on the Incidents page.

Related Documentation

- [Using the Salesforce Adapter with Oracle Integration 3](#)
- [Using the ServiceNow Adapter with Oracle Integration 3](#)

Sync Salesforce Contacts with ServiceNow Users

Use this recipe to synchronize Salesforce contacts with ServiceNow users.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates a user in ServiceNow when a new contact is created in Salesforce. Also, whenever a contact in Salesforce is updated, the recipe updates the corresponding user in ServiceNow.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a Salesforce contact is created or updated, the integration flow of the recipe is triggered, and it creates or updates the corresponding user in ServiceNow. Basic data such as first name, last name, state, email, and so on are synchronized between Salesforce and ServiceNow platforms.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Salesforce
- ServiceNow
- An account on Salesforce with the Administrator role
- An account on ServiceNow with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Salesforce and ServiceNow instances in order to successfully connect to these external systems using Oracle Integration and synchronize Salesforce contacts with ServiceNow users.

Configure Salesforce

To access Salesforce using Oracle Integration and retrieve data about Salesforce contacts that are created or updated, you must perform certain configurations 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.

Also, you must identify your Salesforce instance type and your current Salesforce API version. In addition, create an outbound message to trigger the integration flow of the recipe, and create a workflow rule to send the outbound message when a Salesforce contact is created or updated. Finally, create a custom field for Salesforce contact records to store ServiceNow user IDs.

Login to your Salesforce instance as an **Administrator** and execute the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Contacts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - 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 username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.

- e. Set a password and note the same.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
4. Identify your Salesforce instance type. See Identify Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
5. Create a custom field for contact records.

This recipe uses unique IDs associated with user records in ServiceNow to synchronize ServiceNow users with Salesforce contacts. Create a custom field for Salesforce contact records to hold the ServiceNow IDs.

- a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Contacts**, and then click **Fields**.
- b. On the Contact Fields page, scroll to the Contact Custom Fields & Relationships section and click **New**.
- c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter ServiceNowUserId 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 Contact Fields page, you can see the new field added under the Account Custom Fields & Relationships section.
6. 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.
 - ii. Expand **Create**, then **Workflow & Approvals**, and then click **Outbound Messages**.

If you're shown the Understanding Workflow page, click **Continue**.

- b. Click **New Outbound Message** in the resulting page.
- c. On the New Outbound Message page:
- i. Select **Contact** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new outbound message:
 - Enter a name for the outbound message. Note that the **Unique Name** field is automatically populated based on the name you enter.
 - Optionally, enter a description for the message.
 - In the **Endpoint URL** field, enter the following URL: `https://<OIC_FQDN:OIC_SSL_PORT>/ic/ws/integration/v1/flows/salesforce/ORCL-R-SFDC_CON_TO_SNOW_USER_SYC/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 **Contact fields to send**, select all the fields and click **Add**.
 - Click **Save**.

The new outbound message is created and displayed on your browser.

7. Now, create a workflow rule to send the outbound message (created previously) when a Salesforce contact is created or updated.
- 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 **Contact** in the **Object** field, and click **Next**.
 - ii. Enter the following details for the new workflow rule:

- Enter the rule's name.
 - Optionally, enter a description for the rule.
 - In the Evaluation Criteria section, select the **created, and every time it's edited** radio button.
 - In the Rule Criteria section, select the **formula evaluates to true** option, and define the following rule criteria: Contact: First NameCONTAINSSF
 - Click **Save & Next**.
- iii. On the next page, click the **Add Workflow Action** button and choose **Select Existing Action**.
- iv. On the Select Existing Actions page:
- Select **Outbound Message** in the **Search** field.
 - In the **Available Actions** box, select the outbound message you created previously, for example, Outbound Message: <name of your outbound message>.
 - Click **Add** to move your selection to the **Selected Actions** box.
 - Click **Save**.
- v. Click **Done**.
- The new workflow rule is created and displayed on your browser.
- d. Click **Activate** to activate the rule.

Configure ServiceNow

You can use admin account credentials or custom account credentials while configuring the ServiceNow connection in Oracle Integration.

If you do not want to use an admin account, create a custom integration user and assign the required role and permissions to the user. For information about creating a custom user, see *Prerequisites for Creating a Connection in Using the ServiceNow Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.

Field	Information to Enter
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .
3. In the Security section, enter the following details:
- | Field | Information to Enter |
|------------------------|--|
| Security Policy | Leave Salesforce Username Password Policy selected. |
| Username | Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce . |
| Password | Enter the password of the account created for Oracle Integration on Salesforce.
Note: To the password, you must also append the security token generated for the same account. |
4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
- A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle ServiceNow Connection

- In the Connections section, click the connection name.
 - In the Connection Properties section, enter the ServiceNow host name. For example: `https://instance_name.servicenow.com`.
 - In the Security section, enter the following details:
- | Field | Information to Enter |
|-----------------|--|
| Username | Enter your ServiceNow username. See Configure ServiceNow . |
| Password | Enter your ServiceNow password. |
- Click **Save**. If prompted, click **Save** again.
 - Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
- A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Update Integration Property

Update the integration property of the recipe's integration flow with an appropriate value.

- In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
- In the Update property values panel, update the following integration property.
EmailTo: This integration property holds the email address to which run-time exception emails are sent. Enter an email address of your choice.

3. Click **Submit**.

A message confirms that the integration properties have been updated successfully.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).

2. Run the recipe.

To run the recipe, you must create an account in Salesforce.

- a. Log in to your Salesforce instance.
- b. On the Salesforce Setup page, click the **Contacts** tab.
- c. On the Contacts Home page, click **New**.
- d. On the New Contact page, enter the account details, such as name, description, and so on. Click **Save**.

You've now created an account record in Salesforce and triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your ServiceNow instance and check if a new company has been created.
 - a. In the left navigation pane, under **Organization**, click **Users**.
 - b. On the Users page, search for the newly created company.
 - c. Click the user record's name to view details. Check if the newly created user's data matches with the Salesforce contact that you created earlier.

Related Documentation

- *Using the Salesforce Adapter with Oracle Integration 3*
- *Using the ServiceNow Adapter with Oracle Integration 3*

Sync Stripe Customers with Salesforce Accounts

Use this recipe to synchronize customers in Stripe with Salesforce accounts.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This is an application-driven recipe that creates an account in Salesforce when a new customer is created in Stripe. Also, whenever a customer in Stripe is updated, the recipe updates the corresponding account in Salesforce.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. When a customer is created or updated in Stripe, the integration

flow of the recipe is triggered, and it creates or updates the corresponding account record in Salesforce.

System and Access Requirements

- Oracle Integration, Version 21.2.1.0.0 or higher
- Stripe
- An account on Stripe with the Administrator role
- Salesforce
- An account on Salesforce with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Stripe and Salesforce instances in order to successfully connect to these external systems using Oracle Integration and synchronize customers and accounts.

Configure Stripe

To access the Stripe instance from Oracle Integration, you must perform certain configurations on your Stripe instance.

1. Register a webhook endpoint. See [Add a webhook endpoint](#).
 - a. In the **Endpoint URL** field, enter your Oracle Integration endpoint URL. For example, `https://<username>:<password>@<host>/ic/api/integration/v2/flows/rest/project/ORCL-R-STIPRE_SFDC_CUST_ACC_SYNC/ORCL-R-STRIPE_SFDC_CUST_ACC_SYNC/1.0/createCustomer`.
 - b. In the **Select events** field, select **customer.created** and **customer.updated**.

Configure Salesforce

To access Salesforce using Oracle Integration, you must perform certain configurations 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 account records.

Log in to your Salesforce instance as an **Administrator** and complete the following tasks.

Note

The steps provided here apply to the Salesforce Classic UI. If you're using the Lightning Experience UI on your Salesforce instance, switch to the Classic UI. 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:
 - i. Scroll to the Administrative Permissions section and ensure that the **API Enabled** check box is selected.
 - ii. Scroll to the Standard Object Permissions section and perform the following actions.
 - In the **Accounts** row, leave the **Read**, **Create**, **Edit**, and **Delete** boxes checked. Additionally, select the **ViewAll** check box.
 - 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 username.
 - iii. In the **User License** field, select **Salesforce**.
 - iv. In the **Profile** field, select the profile you created previously, that is, **API Enabled**.
 - v. Scroll to the end of the page, ensure that the **Generate new password and notify user immediately** check box is selected, and click **Save**.

The user account is now created, and a verification email is sent to the email address you provided for the account.
- d. Log in to the corresponding email account and click the **Verify Account** button in the email message from Salesforce. You're redirected to the Salesforce instance to set a password for the new user account.

- e. Set a password and note it down.

Subsequently, you're signed in to the Salesforce instance with the new account.

Note

If you're shown the Lightning 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. See Find Your Current Salesforce API Version in *Using the Salesforce Adapter with Oracle Integration 3*.
- 4. Identify your Salesforce instance type. See Identify Instance Type of Your Salesforce Organization in *Using the Salesforce Adapter with Oracle Integration 3*.
- 5. Create a custom field for account records.

This recipe uses unique IDs associated with customer records in Stripe to synchronize Stripe customers with accounts in Salesforce. Create a custom field for Salesforce account records to hold the Stripe IDs.

- a. On the Salesforce Setup page:
 - i. Scroll to the Build section in the left navigation pane.
 - ii. Expand **Customize**, then **Accounts**, and then click **Fields**.
- b. On the Account Fields page, scroll to the Account Custom Fields & Relationships section and click **New**.
- c. On the New Custom Field page:
 - i. Find and select the **Text** radio button, and click **Next**.
 - ii. Enter the following details for the new custom field:
 - Enter Account_Ext_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 Accounts Fields page, you can see the new field added under the Account Custom Fields & Relationships section.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Stripe REST Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave OAuth2.0 Or Basic Authentication .

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

Configure the Oracle Salesforce Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Select Salesforce.com Instance Type	Select Production or Sandbox based on your Salesforce instance type.
API Version	Enter your current Salesforce API version. To obtain the API version, see Configure Salesforce .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave Salesforce Username Password Policy selected.
Username	Enter the username of the account created for Oracle Integration on Salesforce. See Configure Salesforce .

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.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
Create a customer in Stripe. See [Create a customer](#).
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if new accounts are created in your Salesforce instance.
 - a. Log in to your Salesforce instance.
 - b. On the Salesforce Setup page (in the Classic UI), click the **Accounts** tab.
 - c. On the Accounts Home page, select **All Accounts** in the View field, and click **Go!**. All the accounts in your Salesforce instance are displayed.

Related Documentation

- *Using the REST Adapter with Oracle Integration 3*
- *Using the Salesforce Adapter with Oracle Integration 3*

Sync Supplier Updates Between Oracle Procurement Cloud and Oracle Autonomous Transaction Processing

Use this recipe to synchronize supplier updates between Oracle Procurement Cloud and Oracle Autonomous Transaction Processing (ATP).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe updates the Oracle Autonomous Transaction Processing (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.

To use the recipe, you must install the recipe and configure the connections and other resources within it. 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 ATP database table. 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 ATP database table, thereby synchronizing the supplier data updates between the systems.

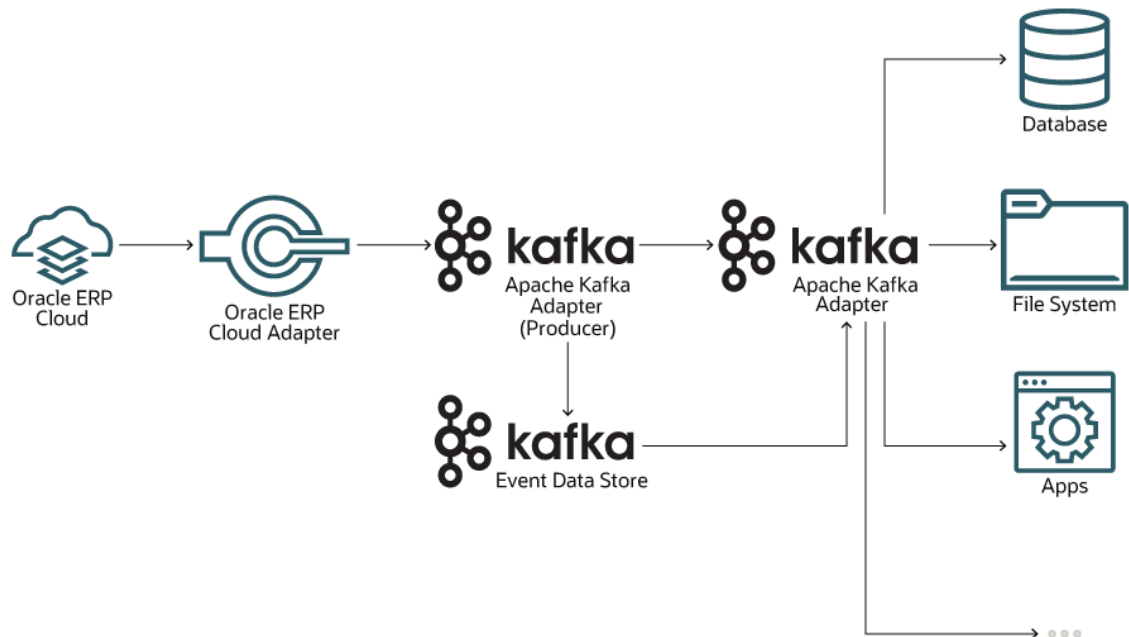
System and Access Requirements

- Oracle Integration, Version 21.2.2.0.0 (210505.1400.40944)
- Oracle ERP Cloud
- Oracle Autonomous Transaction Processing
- Apache Kafka
- An account on Oracle ERP Cloud with the Administrator and Procurement Manager roles
- An account on Apache Kafka with the Administrator role
- An account in Oracle Autonomous Transaction Processing with the Administrator role
- An 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 ATP database table.



Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud, Apache Kafka, and Oracle Autonomous Transaction Processing instances in order to successfully connect to these external systems using Oracle Integration and achieve synchronization.

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.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles/privileges to the user account. See *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - AttachmentsUser
 - FND_MANAGE_CATALOG_SERVICE_PRIV
 - ASM_IMPLEMENTATION_MANAGER_DUTY

3. 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.

- a. Log in to your Oracle ERP Cloud instance.

- b. Click **My Enterprise** from the navigator on the home page, and then click **Feature Updates**.
- c. In the Features Overview page, select the **Available Features** tab.
- d. Enter Suppliers in the functional area search field.
All available features that belong to the suppliers functional area get listed.
- e. In the row for the feature Enabled Outbound Supplier Profile Integration Using Oracle Cloud, click **Enabled**.
- f. In the resulting window, select the **Enable** check box for the feature Enable Outbound Supplier Profile Integration Using Oracle Cloud.
- g. Click **Done**.

Configure Apache Kafka

Complete certain configuration tasks to successfully connect to Apache Kafka from Oracle Integration and achieve synchronization.

1. Get the Apache Kafka Cluster Bootstrap Server URL. See Prerequisites for Creating a Connection in *Using the Apache Kafka Adapter with Oracle Integration 3*.
2. 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 in *Using Integrations in Oracle Integration 3*.
- Connectivity Agent Restrictions in *Using Integrations in Oracle Integration 3*

To install and configure the connectivity agent:

- a. Create an agent group.
- b. Download and install the connectivity agent.
3. Configure an Apache Kafka topic.
 - a. 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.

- b. 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

Complete the following configuration tasks on your Oracle Autonomous Transaction Processing instance.

The following steps give an overview of the tasks you need to perform in order to create a connection from Oracle Integration to Oracle Autonomous Transaction Processing. For detail information, see *Prerequisites for Creating a Connection in Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*.

1. Download the client credentials wallet.

This is a zip file containing the client security credentials. By default the file name is `Wallet_<dbname>.zip`. You can save this file as any file name you want.

2. 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 Autonomous Transaction Processing connection security from Oracle Integration.

In addition to the above configuration tasks, you need to set up a database table in your Oracle Autonomous Transaction Processing instance to synchronize supplier information.

Create an ATP database table

1. Log in to your Oracle Autonomous 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**.
5. 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
<code>v_SupplierId</code>	number	Yes	Yes
<code>v_SupplierName</code>	varchar2(20)	-	-
<code>v_SupplierNumber</code>	number	-	-
<code>v_BusinessRelationship</code>	varchar2(20)	-	-
<code>v_Status</code>	varchar2(20)	-	-
<code>v_DUNSNumber</code>	int	-	-
<code>v_TaxRegistrationNumber</code>	varchar2(20)	-	-
<code>v_TaxpayerId</code>	varchar2(20)	-	-
<code>v_TaxpayerCountryCode</code>	varchar2(20)	-	-
<code>v_SupplierTypeCode</code>	varchar2(20)	-	-
<code>v_TaxOrganizationType</code>	varchar2(20)	-	-

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the username of the account created for Oracle Integration on Oracle ERP Cloud. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on Oracle ERP Cloud.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Apache Kafka Connection

1. In the Connections section, click the connection name.
2. In the **Bootstrap Servers** field, enter the Bootstrap Server URL you obtained earlier while configuring Apache Kafka. See [Configure Apache Kafka](#).
3. In the **Security Policy** field, select **No Security Policy**.
4. In the Agent Group section, select the agent group.
 - 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 [Configure Apache Kafka](#).
 - c. Click **Use**.
5. Click **Save**. If prompted, click **Save** again.
6. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

7. To return to the project workspace, click **Go back** .

Configure the Oracle ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Service Name of your Oracle Autonomous Transaction Processing instance. See [Configure Oracle Autonomous Transaction Processing](#).
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC Over SSL .
Wallet	Click Upload to upload the credential wallet file that you obtained. See Configure Oracle Autonomous Transaction Processing .
Wallet Password	Enter the password for your credential wallet file.
Database Service Username	Enter your Oracle Autonomous Transaction Processing account username.
Database Service Password	Enter your Oracle Autonomous Transaction Processing account password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Lookup Table

Edit the lookup table and enter an email ID to receive error messages.

1. In the Lookups section, click the lookup name.
2. In the **Property_Value** column, enter an email ID of your choice.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to 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 successfully triggered the recipe.

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to the Oracle Autonomous Transaction Processing 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 3*
- *Using the Apache Kafka Adapter with Oracle Integration 3*
- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*

Sync Work Schedules between Workday and Oracle Field Service Cloud

Use this recipe to synchronize work schedules between Workday and Oracle Field Service Cloud.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes work schedules of employees between Workday and Oracle Field Service Cloud according to a schedule specified in Oracle Integration.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

When triggered, the integration flow **Oracle FTP OFS Employee2Resource Sync** fetches the Workday employee records from the sFTP to Oracle Field Service Cloud. Subsequently, the **Oracle Workday OFS Resource Work Schedule Sync** integration flow synchronizes the work schedules between Workday and Oracle Field Service Cloud.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Workday
- An account on Workday with the Administrator role
- A secure FTP (SFTP) server or File Server
- An FTP client to access the SFTP server
- Oracle Field Service Cloud
- An account on Oracle Field Service Cloud with the Administrator role
- Employee records exported from Workday to SFTP. See [Export Workday Employees to an FTP Server](#).

Before You Install the Recipe

You must perform the following configuration tasks on your SFTP, Workday, and Oracle Field Service Cloud instances in order to successfully connect to these external systems using Oracle Integration and synchronize work schedules.

Access Your FTP Server

Obtain a SFTP server and ensure that you have noted the following:

1. Username and password to connect to the SFTP server.
2. Host name or IP address and port number of the SFTP server.
3. Name of the file from which the Workday employee records must be fetched.
4. Path of the file from which the Workday employee records must be fetched.

Also, ensure that you have the read and write permissions on the SFTP server directory to and from which you want to transfer files.

Configure Workday

To access Workday using Oracle Integration and sync work schedules, you must perform certain configuration tasks on your Workday instance.

For general configuration tasks, see Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.

In addition, you must create a RaaS report to fetch the work schedules of employees from Workday and export it as a web service. And, before you create RaaS reports you must create calculated fields for the following items:

- Start Date
- End Date
- Work Schedule Calendar

Note

In your Workday instance, ensure that work schedules are assigned to employees, and these schedules are updated before running the recipe. Only updated schedules are picked up for synchronization.

1. Create calculated fields.
 - a. In the quick find box, search for **Create Calculated Fields** task.
 - b. In the pop-up window, enter the **Field name**. For example: CF_Test. Also, select the **Business Object** as **Worker Business Process**, and **Function** as **Lookup Related Value**.
 - c. Click **OK**. You are redirected to Create Calculated Field - Lookup Related Value page.
 - d. Under Calculation, select the **Lookup Field** name as **Work Schedule Assignment Event** and **Return Value** as **Start Date**.

Repeat the steps to create calculated fields for **End Date** and **Work Schedule Calendar – Proposed**.
2. Create a RaaS report.
 - a. Log in to the Workday instance.
 - b. In the quick find box, search for **Create custom report**.
 - c. In the Create Custom Report window:
 - i. Enter the **Report Name**. For example: RPT Business process transaction schedule.
 - ii. Select the **Report Type** as **Advanced**.
 - iii. Check the **Enable As Web Service** box.
 - iv. Uncheck the **Optimized for Performance** box.
 - v. Select the **Data Source** as **Business Process Transactions**.
 - vi. Click **OK**. You are redirected to Edit Custom Report page.
 - vii. Add the **Fields** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.
 - viii. Click **OK**.
3. After creating custom reports, you can specify how the report results should be sorted.
4. In the Prompts section, check the **Populate Undefined Prompt Defaults** box.

Note

The data source **Business Process Transactions** uses built-in prompts such as **Effective Date**, **Entry Date**, and so on. Based on the report you have chosen, Workday prompts you to fill in specific reporting criteria.

Configure Oracle Field Service Cloud

Retrieve the client ID and secret from Oracle Field Service Cloud by adding an application.

To add an application:

1. Log in to an Oracle Field Service Cloud instance.
2. In the top-left corner, click the hamburger icon, then click **Configuration**.
3. On the Configuration page, under **Subsystems**, click **Applications**.
4. Click **Add Application**.
5. Enter the **Application Name** and **Application ID**, give the required API access. See [Authorize Access to APIs](#).
6. Click **Submit**.
7. Select the type of token service for the registered client application from the **Token service** drop-down list.
By default, the type of token service is **Oracle Field Service Cloud**.
8. Select the authentication settings as **Authenticate using Client ID/Client Secret**.
9. Click **Show Client ID and Client secret** to copy them for configuring the connection in Oracle Integration.
10. Click **Save**.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle FTP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Address	Enter the host address of your SFTP server. See Access Your FTP Server .
FTP Server Port	Enter 22.
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy .
Username	Enter the user name to connect to your SFTP server. See Access Your FTP Server .
Password	Enter the password to connect to your SFTP server.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Workday Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Host Name	Enter the Workday host name.
Tenant Name	Enter the tenant name of the account created for Oracle Integration on Workday. See Configure Workday .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Workday Username Token Policy .
Workday Integration User	Enter the user name of the account created for Oracle Integration on Workday. See Configure Workday .
Password	Enter the password of the account created for Oracle Integration on Workday.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle Field Service Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Field Service Cloud API URL	Enter the URL of your Oracle Field Service Cloud instance. See Configure Oracle Field Service Cloud .
Instance ID	Enter the ID of your Oracle Field Service Cloud instance. For example, ofsc-xyz.test.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Client Credentials .
Client Id	Enter the client ID obtained from your Oracle Field Service Cloud instance. See Configure Oracle Field Service Cloud .
Client Secret	Enter the client secret obtained from your Oracle Field Service Cloud instance.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Lookup Tables

The recipe contains the following four lookup tables with default mappings. Edit them as necessary.

- Oracle_FTP_OFS_WorkerStatus**: Used to map **Workday Employee Status** with **OFSC Resource Status**.
- Oracle_FTP_OFS_WorkerType**: Used to map **FTP Worker Type** with **OFSC Resource Type**.
- Oracle_Workday_OFS_RecordType**: Used to map **Workday Calendar** with **OFSC Schedule Label**.
- Oracle_Workday_OFS_WorkSchedule**: Used to map **Workday Schedule** with **OFSC Schedule**.

- In the Lookups section, click the lookup name.
- Edit the lookup keys in the lookup table.
- Click **Save**. If prompted, click **Save** again.
- To return to the project workspace, click **Go back** .

Update the Report Owner ID

If the RaaS report owner is changed in Workday, you must configure the Workday Adapter in Oracle Integration with the updated report-owner ID.

Note

Before updating the RaaS Report Owner ID, ensure that you deactivate the **Oracle Workday OFS Resource Work Schedule Sync** integration.

To update the RaaS report owner ID:

- In the Integrations section of the project workspace, click **Actions**  on the **Oracle Workday OFS Resource Work Schedule Sync** integration, then select **Edit** .

2. On the integration canvas, select **ExportWorkSchedule**, and then click **Edit** . The Adapter Endpoint Configuration wizard appears.
3. Click **Next** until you reach RaaS Services page.
4. In the RaaS Services page, verify if the new report-owned ID has been automatically updated.
5. Click **Done**.

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update integration property values for the **Oracle FTP OFS Employee Resource Sync** integration flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the integration properties with appropriate values.

Integration Property	Description
Default Employee Status	Specify the default value of Workday employee status.
EmailTo	Specify the email address to which run-time exception emails are sent. Enter an email address of your choice.
DefaultResourceType	Specify the default value of resource type.
RetryCount	Specify the number of possible retries if any record fails while creating resources at run time.

- c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.
3. Update integration property for the **Oracle Workday OFS Resource Work Schedule Sync** integration flow.
 - a. In the Integrations section, click **Actions**  on the integration flow, then select **Update property values**.
 - b. In the Update property values panel, update the following integration property with appropriate value.

EmailTo: Specify the email address to which run-time exception emails are sent. Enter an email address of your choice.
 - c. Click **Submit**.

A message confirms that the integration properties have been updated successfully.
4. Run the integration **Oracle FTP OFS Employee2Resource Sync** to fetch the resources from FTP to Oracle Field Service Cloud.
 - a. In the Integrations section of the project workspace, click **Actions**  on the integration flow, then select **Run**.

The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

- **schedule_paramErrorFilePath:** Specify the FTP error file path in which the error records are stored after unsuccessful execution.
 - **schedule_param_ErrorFileName:** Specify the name of the FTP error file in which the error records is stored after unsuccessful execution.
 - **schedule_param_FileName:** Specify the name for the FTP file from which the Workday employee records must be fetched.
 - **schedule_param_FilePath:** Specify the path of the FTP file from which the Workday employee records must be fetched.
 - **schedule_param_NextRun:** Stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Workday employee records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm.
- b. On the Configure and run page, click **Run**.
- You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
6. Log in to your Oracle Field Service Cloud instance and check for the new resources created.
 - a. Log in to the Oracle Field Service Cloud instance.
 - b. In the top-left corner, click the hamburger icon, then click **Resources**.
 - c. In the search box, enter the ID of the Workday employee you want to search for.

Note

The Workday employee ID is the sourceSystemId in FTP.

- d. Check if the new resource is created.
7. Run the integration **Oracle Workday OFS Resource Work Schedule Sync** to synchronize the work schedules with Oracle Field Service Cloud.
 - a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

The Configure and run page is displayed with the following parameter. Update the parameter with appropriate value.

- **schedule_para_LastRunTime:** Stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Workday schedules created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

- You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.
- If any record fails while setting up the work schedule in Oracle Field Service Cloud, an email notification is sent to the user. The admin must update that record in Workday again, so that it is fetched in the next run automatically.

8. Monitor the execution of the integration flow in Oracle Integration. See Step 4.
9. Log in to your Oracle Field Service Cloud instance and check for work schedules assigned to the resources.
- a. Log in to the Oracle Field Service Cloud instance.
- b. In the top-left corner, click the hamburger icon, then click **Resources**.
- c. In the search box, enter the ID of the Workday employee you want to search for.

Note

The Workday employee ID is the `sourceSystemId` in FTP.

- d. To view the assigned work schedule, open the resource, then click **Resource Calendar**.

Related Documentation

- *Using the Workday Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Oracle Field Service Cloud Adapter with Oracle Integration 3*

Sync Workday Employees with ServiceNow Users

Use this recipe to synchronize employees in Workday with ServiceNow users.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes Workday employees with ServiceNow users on a scheduled basis. It creates or modifies user records in ServiceNow according to the employee data received from

Workday. In addition, if a ServiceNow user creation fails, the recipe creates a corresponding ticket in ServiceNow.

To use the recipe, you must install the recipe and configure the connections and other resources within it. Subsequently, you can activate and run the integration flow of the recipe manually or specify an execution schedule for it.

The scheduled integration **Oracle Workday ServiceNow Employee Sync** reads the employee records from Workday and creates or modifies corresponding users in ServiceNow. Subsequently, if any ServiceNow user creation fails, the scheduled integration **Oracle Workday ServiceNow Emp Onboarding Ticket** reads the failed records and creates or updates corresponding tickets in ServiceNow. The app-driven integration **Oracle REST Workday ServiceNow** is triggered by each of the scheduled integration flows for their respective tasks.

Basic employee data such as first name, last name, email, and so on are synchronized between Workday and ServiceNow.

System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Workday
- An account on Workday with the Administrator role
- ServiceNow
- An account on ServiceNow with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Workday and ServiceNow instances in order to successfully connect to these external systems using Oracle Integration and synchronize employees and users.

Configure Workday

To access Workday using Oracle Integration and sync employees, you must perform certain configuration tasks on your Workday instance.

For general configuration tasks, see Prerequisites for Creating a Connection in *Using the Workday Adapter with Oracle Integration 3*.

In addition, you must create an Enterprise Interface Builder (EIB) report to import and export Workday data in bulk. And, before you create an EIB report, you must create a custom report with custom fields, calculated fields, and business objects.

1. Create custom fields.
 - a. In Workday, obtain access to **Create Custom Object**.
 - b. Select **Workday Object** as **Worker (Effective Dated)** and **Custom object Name** as **SN_TicketId**.
 - c. Select the values as below.
 - **General Settings**

Custom Object Name	Custom Object Name	SN_TicketId

Web Service Alias	ID	snTicketId
Allow Multiple Instances per Object?		No

• **Field Definitions**

Current Fields	Field Label	SN_TicketId
	Field Type	Text

• **Permissions**

Security	Domain	Person Data: Personal Data, Person Data: Personal Information

• **Edit Custom Fields**

Edit Custom Fields	Display Value	SN_TicketId

- d. In Workday, search for **Edit Worker additional data business process**.
- e. In the **Edit worker additional data business process**, go to **Action** and click **Edit definition**.
- f. Add a new step and select values as below.

Note

In **Group** column, select the **Security Group** which should be allowed to edit the custom field.

Step	Order	Type	Specify	Option al	Group	All	Run as User	Due Date	Due Data Is Based On Effective Date	Compl ete
	a5	Edit Additio nal Data	SN_Ti cketId	No	HR Admini strator					

- g. Click **OK**.
- Similarly, create custom fields to store **Ticket_ErrorMessage**, **User_ErrorMessage** and **ServiceNowUserId**.

Note

Provide **Get access of Person Data: Personal Data, Person Data: Personal Information to ISU user** being used for getting data from custom report.

- 2. Create calculated fields.
 - a. In the quick find box, search for **Create Calculated Fields** task.

- b. In the pop-up window, enter the **Field name**. For example: CF_new_hire_or_updated. Also, select the **Business Object** as **Worker**, and **Function** as **Evaluate Expression**.
- c. Click **OK**. You are redirected to **Create Calculated Field - Evaluate Expression** page.
- d. Under Calculation, select the **Field Type** as **Text** and **Default Value** as **Empty String**.
- e. In the Condition column, create the following calculated fields:

- **CF-T/F_oic_updated**

And/Or	(Field	Calculated Field Details	Operator	Comparison Type	Comparison Value
And	CF_Legal_name_change	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_legal_name_change? Sort Field: Effective Date Sort Direction: Descending (Z to A) Instance to be Returned: First Occurrence Function: True or False	is not empty	NA	NA
Or	CF_new_event_exist	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_effective date range Fields To Aggregate: Business Process Event Function: True or False	is not empty	NA	NA
Or	(CF_usererrormsg	Business Object: Worker Lookup Field: Worker Additional Data Related Business Object: Worker Custom Data Snapshot Return Value: User_ErrorMessage Function: True or False	equal to	Value specified in this filter	User failed
And	CF_user_ticket_error	Business Object: Worker Lookup Field: Worker Additional Data Related Business Object: Worker Custom Data Snapshot Return Value: Ticket_ErrorMessage Function: True or False	equal to	Value specified in this filter	Ticket failed)

- **CF_legal_name_change?**

And/O r	(Field	Calculated Fields Details	Operator	Compariso n Type	Compariso n Value
And	Business Process Type	Business Object: Event Function: True or False	in the selection list	Value specified in this filter	Legal Name Change
And	Status	Business Object: Event Function: True or False	greater than or equal to	Value specified in this filter	Successfull y Completed
And	Effective Date	Business Object: Action Event Function: True or False	greater than or equal to	Value from another field	Prompt- Start Date

- **CF_oic_is_new_hired?**

And/O r	(Field	Calculated Fields Details	Operator	Compariso n Type	Compariso n Value
And	Hire Date	NA	greater than or equal to	Value from another field	Prompt- Start Date
And	CF_Legal_ name_ change	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_legal_name_change ? Sort Field: Effective Date Sort Direction: Descending (Z to A) Instance to be Returned: First Occurrence Function: True or False	is empty	NA	NA
And	CF_new_e vent_exist	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_effective date range Fields To Aggregate: Business Process Event Function: True or False	is empty	NA	NA

- f. In the **Return Value If Condition is True** column, create the following calculated fields.

Calculated Field	Details
CF_oic_updated	Business Object: Worker Text Constant: Updated

Calculated Field	Details
CF_oic_NEW_HIRED	Business Object: Worker Text Constant: NEW HIRE

Configuration details for other related calculated fields are listed below.

Calculated Field	Details
CF_legal_name_change	Business Object: Worker Source Field: Worker History Related Business Object: Action Event Condition: CF_legal_name_change? Sort Field: Effective Date Sort Direction: Descending (Z to A) Instance to be Returned: First occurrence Function: True or False
CF_Ticket_ErrorMessage?	Business Object: Worker Function: Lookup Related Value Lookup Field: Worker Additional Data Related Business Object: Worker Custom Data Snapshot Return Value: Ticket_ErrorMessage
CF_work_email	Business Object: Worker Function: Lookup Value As Of Date Source Field: Email – Work Data As Of: Effective Date: empty Data Entry Date: Entry Date: CF_Last_Functionally_updated_-1millisecond
CF_Last_Functionally_updated_-1mili second	Business Object: Global Function: Increment or Decrement Date Date Field: Last Functionally Updated Milliseconds to Add or Subtract: -1 Return Blank Date on Error: Yes
CF_proposed_Supervisory Organization	Business Object: Worker Function: Lookup Value As Of Date Source Field: Supervisory Organization Data As Of: Effective Date: empty Data Entry Date: Entry Date: CF_Last_Functionally_updated_-1mili second
CF_proposed_Phone - Primary Work	Business Object: Worker Function: Lookup Value As Of Date Date Source Field: Phone - Primary Work Data As Of: Effective Date: empty Data Entry Date: Entry Date: CF_Last_Functionally_updated_-1mili second

Calculated Field	Details
CF_User_ErrorMessage?	Business Object: Worker Function: Lookup Related Value Lookup Field: Worker Additional Data Related Business Object: Worker Custom Data Snapshot Return Value: User_ErrorMessage
CF_previous_business_title	Business Object: Worker Function: Lookup Value As Of Date Source Field: Business Title Data As Of: Effective Date: empty Data Entry Date: Entry Date:CF_Last_Functionally_updated_-1mili second

Also, configure the following calculated fields with the following conditions.

- **CF_effective date range**

And/Or	(Field	Calculated Field	Operator	Comparison Type	Comparison Value	)
And	Business Processes Definition	Business Object: Action Event Function: True or False	in the selection list	Value specified in this filter	• Change Job (Default Definition) • Change Job for Global Modern Services • Change Job for IT Services Group • Change Job for Sales • Change Organization Assignments for Worker (Default Definition) • Change Organization Assignments for Worker for Field Sales - North America Group • Change Organization Assignments for Worker for Global Modern Services • Change Organization Assignments for Worker for Marketing • Change Organization Assignments for Workers by Organization (Default Definition) • Change Organization Assignments for Workers by Organization for Global Modern Services • Contact Change (Default Definition) • Edit Worker Additional Data Event (Default Definition) • Edit Worker Additional Data Event for Global Modern Services • Move Worker (Supervisory) (Default Definition) • Move Worker (Supervisory) for Global Modern Services • Move Workers (By Organization) (Default Definition) • Move Workers (By Organization) for Global Modern Services	)

And/Or	(Field	Calculated Field	Operator	Comparison Type	Comparison Value	)
					• Move Workers (Supervisory) (Default Definition) • Move Workers (Supervisory) for Global Modern Services • Personal Information Change (Default Definition) • Personal Information Change for Global Modern Services • Switch Primary Job (Default Definition) • Switch Primary Job for Global Modern Services • Title Change (Default Definition) • Title Change for Global Modern Services • Work Contact Change (Default Definition)	
And	Status	Business Object: Action Event Function: True or False	equal to	Value specified in this filter	Successfully Completed	
And	Effective Date	Business Object: Action Event Function: True or False	greater than or equal to	Value from another field	Prompt - Start Date	

3. Create a custom report.
 - a. Log in to the Workday instance.
 - b. In the quick find box, search for **Create custom report**.
 - c. In the Create Custom Report window:
 - i. Enter the **Report Name**. For example: RPT_new_hire_report.
 - ii. Select the **Report Type** as **Advanced**.
 - iii. Check the **Enable As Web Service** box.
 - iv. Uncheck the **Optimized for Performance** box.
 - v. Select the **Data Source** as **Active and Terminated Workers**.
 - vi. Click **OK**. You are redirected to Edit Custom Report page.
 - vii. Under the **Columns** tab, add the **Fields** and **Custom Fields** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

Business Object	Fields	Column Heading Override XML Alias
Worker	Employee ID	Employee_ID
Worker	First Name	firstName
Worker	Last Name	lastName
Worker	Email - Work	Email_-_Work
Worker	Hire Date	Hire_Date
Worker	Worker Type	Worker_Type
Worker	Business Title	businessTitle
Worker	Supervisory Organization	Supervisory_Organization

Note

Choose **Column Heading Override** as **Department**.

Worker	Phone - Primary Work	primaryWorkPhone
Worker	CF_new_hire_or_updated	CF_new_hire_or_updated
Worker Additional Data	ServiceNowUserId	servicenowuserid
Worker Additional Data	SN_TicketId	snTicketid
Worker Additional Data	User_ErrorMessage	userErrorMessage
Worker Additional Data	Ticket_ErrorMessage	ticketErrorMessage

- d. Click **OK**.
- e. In Group column heading section, select all business object as below. (Group Column Heading for each business object will be blank)

Group Column Headings

Business Object	Group Column Heading	Group Column Heading XML Alias
Worker Additional Data		Worker_Additional_Data_group

- f. After creating custom reports, you can specify how the report results should be sorted.
- g. Under the **Filter** tab, add the **Fields** and **Custom Fields** required for the Workday employee details along with the calculated fields with the relevant **Business Object**.

And/Or	(Fields	Operator	Comparison Type	Comparison Value	)
And	Terminated (based on report date)	equal to	Value specified in this filter	NA	
And	(Hire Date	greater than or equal to	Prompt the user for the value	Default Prompt	
Or	CF_Legal_name _change	is not empty	NA	NA	
Or	(CF_new_event_ exist	is not empty	NA	NA	

And/Or	(Fields	Operator	Comparison Type	Comparison Value	)
And	(Email - Work	none in the selection list	Value from another field	CF_work_email	)
Or	CF_Ticket_Error Message?	equal to	Value specified in this filter	ticket failed	)
Or	CF_User_Error Message?	equal to	Value specified in this filter	user failed	)
Or	Business Title	not equal to	Value from another field	CF_previous_business_title	)
Or	Supervisory Organization	not in the selection list	Value from another field	CF_proposed_Supervisory Organization	)
Or	Phone - Primary Work	not equal to	Value from another field	CF_proposed_Phone - Primary Work	)

- h. In the Prompts section, check the **Populate Undefined Prompt Defaults** box.

Note

The data source **Active and Terminated Workers** uses built-in prompts such as **Start Date**, **End Date**, and so on. Based on the report you have chosen, Workday prompts you to fill in specific reporting criteria.

4. Create an EIB report.
 - a. Log in to the Workday instance.
 - b. In the quick find box, search for **Create EIB** task.
 - c. Enter the **Name** of the EIB task (**EIB_new_hire_report**).
 - d. In the **Create EIB** pop-up, choose **Outbound** and click **OK**.
 - e. In the **General Settings**, enter comments and then click **Next**.
 - f. Under the **Get Data** tab, select the **Data Source Type** as **Custom Report**.
 - g. Select the **Custom Report** as **RPT_new_hire_report** and then click **Next**.
 - h. Under the **Transform** tab, select the **Transformation Type** as **New Custom Report Transformation**, then click **Next**.
 - i. Under the **Deliver** tab, perform the following steps.
 - i. Select the **Delivery Method** as **Workday Attachment**.
 - ii. Enter the **File Name** with the relevant file extension. For example, output.csv.
 - iii. Specify the **Document Retention Policy (in Days)** as **should be less than 180**, and then click **Next**.
 - iv. In the **Details** section, select the **MIME Type** as **CSV**.
 - v. Click **Next**.
 - j. Under the **Review and Submit** tab, verify the details, then click **OK**.
 - k. Once you create the EIB report, the **View Integration System** window appears. In the top-left corner, click the ellipsis next to the EIB task name, choose **Enterprise Interface**, and then select **Configure Transformation**.

- l.** In the **Configure Transformation** window, update the report transformation details, click **OK**, and then click **Done**.
- m.** In the **View Integration System** window, in the top-left corner, click the ellipsis next to the EIB task name, choose **Integration**, and then select **Launch/Schedule**.
- n.** In the **Launch/Schedule Integration** pop-up, select the **Run Frequency** as **Run Now**, and then click **OK**.
- o.** In the **Schedule an Integration** window, specify the **Start Date** and **Hire Date**, and then click **OK**.

The EIB report gets generated.

Configure ServiceNow

You can use admin account credentials or custom account credentials while configuring the ServiceNow connection in Oracle Integration.

If you do not want to use an admin account, create a custom integration user and assign the required role and permissions to the user. For information about creating a custom user, see *Prerequisites for Creating a Connection in Using the ServiceNow Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST API Base URL .
Connection URL	Enter your Oracle Integration instance URL. For instance, <code>https://<instance-name>.com/</code> .

- 3.** In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Basic Authentication .
Username	Enter the username of your Oracle Integration account.
Password	Enter the password of your Oracle Integration account.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle Workday Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:

Field	Information to Enter
Hostname	Enter the Workday host name. For example: <code>https://wd2-impl-services1.workday.com</code> .
Tenant Name	Enter the tenant name of the account created for Oracle Integration on Workday. See Configure Workday .

- In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Workday Username Token Policy .
Workday Integration User	Enter the user name of the account created for Oracle Integration on Workday. See Configure Workday .
Password	Enter the password of the account created for Oracle Integration on Workday.

- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Oracle ServiceNow Connection

- In the Connections section, click the connection name.
- In the Properties section, enter the following details:
 - In the **ServiceNow Instance Name** field, enter your ServiceNow instance URL. For example, `https://instance_name.service-now.com`.
- In the Security section, enter the following details:
 - In the **Username** and **Password** fields, enter your ServiceNow username and password.
- Click **Save**. If prompted, click **Save** again.
- Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

Configure the Lookup Table

The recipe contains the following two lookup tables. Edit them as necessary.

- **ORACLE-BRT-LOOPCOUNT_LOOKUP:** Use this lookup to configure the loop count, Workday EIB report name, and employee types.

Property	Description	Value
LoopCount	Specify the maximum number of iterations for a loop.	10
WorkdayReportName	Specify the name of the Workday EIB report.	EIB_new_hire_report
EmployeeType	Specify the type of the employee.	Employee_ID

- **ORACLE-BRT-WD_SERVICENOW_LOOKUP:** Use this lookup to map the priority and department between Workday and ServiceNow.

Property	Description	Value	WD_Value	SN_Value
Priority	Priority Values mapped between Workday and ServiceNow.	2-High	Sales	Sales

1. In the Lookups section, click the lookup name.
2. In the **Value** column, enter or map the appropriate values for the properties.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Update property values for the **Oracle Workday ServiceNow Employee Sync** integration flow. You can update the integration properties only after activating this flow.
 - a. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
 - b. In the Update Property Values dialog, update the following properties

Integration Property	Description
EmailTo	Specify the email address to which run-time exception emails are sent. Enter an email address of your choice.
RetryCount	Specify the number of possible retries if any record fails while creating resources at run time.
workdayTimeZoneDuration	This integration property converts Oracle Integration time zone to Workday tenant time zone. User can configure this value.

- c. Click **Submit**.

A message confirms that the integration property has been updated successfully.

3. Run the integration **Oracle Workday ServiceNow Employee Sync** to create or modify users in ServiceNow according to the employee data received from Workday. This integration triggers the **Oracle REST Workday ServiceNow** integration.

- a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

The Configure and run page is displayed, where you can specify a value for the `LastRunTime` parameter.

The `LastRunTime` parameter stores the date and time of the most-recent successful run of the integration flow. The parameter's value is automatically updated after each successful run, and only the Workday employee records created after the date-time stamp stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date-time stamp for a specific scenario, enter the date and time of your choice in the **New Value** field in the format, yyyy-MM-dd HH:mm. To fetch all the employee records from the beginning, set the value as zero.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

 Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

4. Run the integration **Oracle Workday ServiceNow Emp Onboarding** to create tickets in ServiceNow for the failed Workday employee records. This integration triggers the **Oracle REST Workday ServiceNow** integration.

- a. In the Integrations section of the project workspace, click **Actions** ... on the integration flow, then select **Run**.

- b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

5. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).

6. Log in to ServiceNow and check for the employee records created or modified and the corresponding tickets created.

- a. View the employee records.

- i. In the **Filter** box, search for **Users**.

- ii. Under **Organization**, select **Users**.

You can see the list of users created or modified.

- b. View the tickets created.

- i. In the **Filter** box, search for **HR Case**.

- ii. Under **All HR Cases**, click **All**.

You can see a list of all the tickets created.

Related Documentation

- *Using the Workday Adapter with Oracle Integration 3*
- *Using the ServiceNow Adapter with Oracle Integration 3*

Sync Opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite Orders

Use this recipe to create a new sales order in Oracle NetSuite when an opportunity is won in Oracle CX Sales and B2B Service.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes opportunities in Oracle CX Sales and B2B Service with Oracle NetSuite sales orders in real time. It uses the standard Oracle CX Sales and B2B Service Adapter and the Oracle NetSuite Adapter. To use the recipe, you must install the recipe and configure the connections and other resources within the recipe.

When an opportunity closes with the status Won in Oracle CX Sales and B2B Service, the main integration flow of the recipe is triggered. From Oracle CX Sales and B2B Service, this integration flow receives the customer and inventory item details associated with the opportunity won. Subsequently, the main flow calls two subsidiary integration flows to fetch the corresponding customer and inventory item details from Oracle NetSuite. The subsidiary flows check if the particular customer and inventory item records exist in Oracle NetSuite. If the records exist, the subsidiary flows return the internal IDs of the records to the main flow. If records don't exist (that is, if it's a new customer or a new inventory item), the subsidiary flows create a new customer record and/or a new inventory item record in Oracle NetSuite and return the internal IDs of these records to the main flow. After it receives the necessary details, the main flow creates a new sales order in Oracle NetSuite, thereby synchronizing the data between Oracle CX Sales and B2B Service and Oracle NetSuite.

Note

- You can run this recipe for one Oracle CX Sales and B2B Service product group at a time.
- For successful execution of the recipe, you must add only one product group to the opportunity in Oracle CX Sales and B2B Service, and you must not add a product item to the opportunity.

System and Access Requirements

- Oracle Integration, Version 21.1.2.0.0 (210129.2200.39447) or higher
- Oracle NetSuite, Version 2018.1.0
- Oracle CX Sales and B2B Service
- An account on Oracle NetSuite with the Administrator role

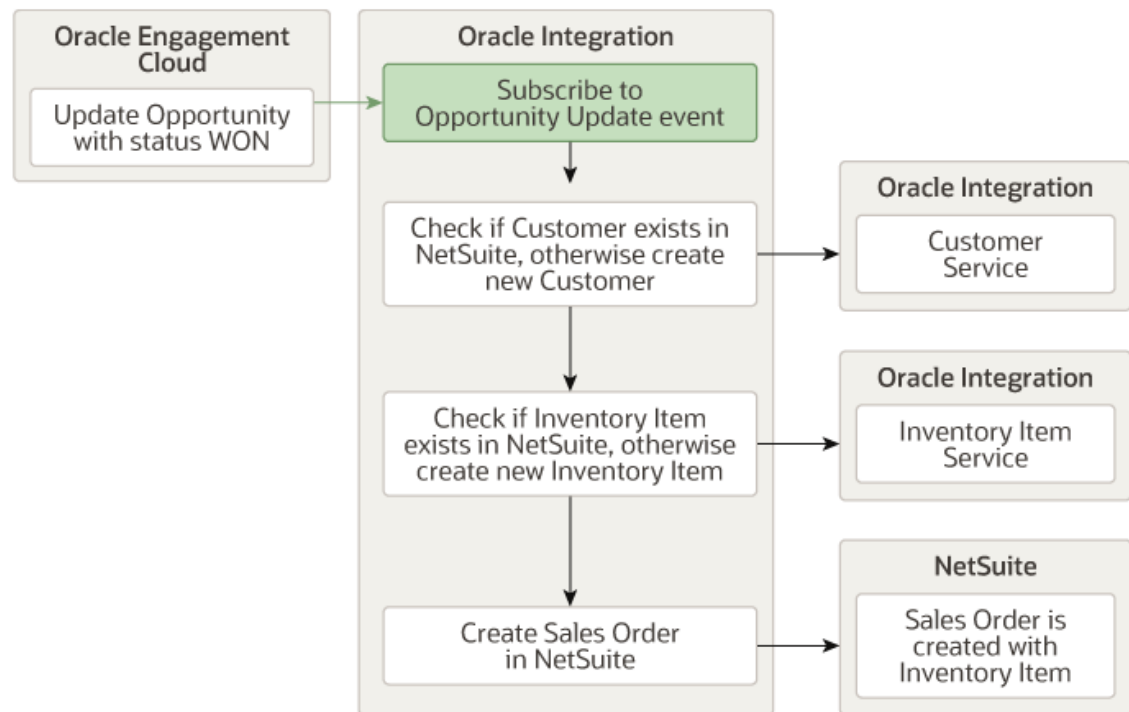
- Accounts on Oracle CX Sales and B2B Service with the Administrator role and the Sales User role (in order to create and close opportunities)

Recipe Schemas

This section describes the integration flows present in the recipe.

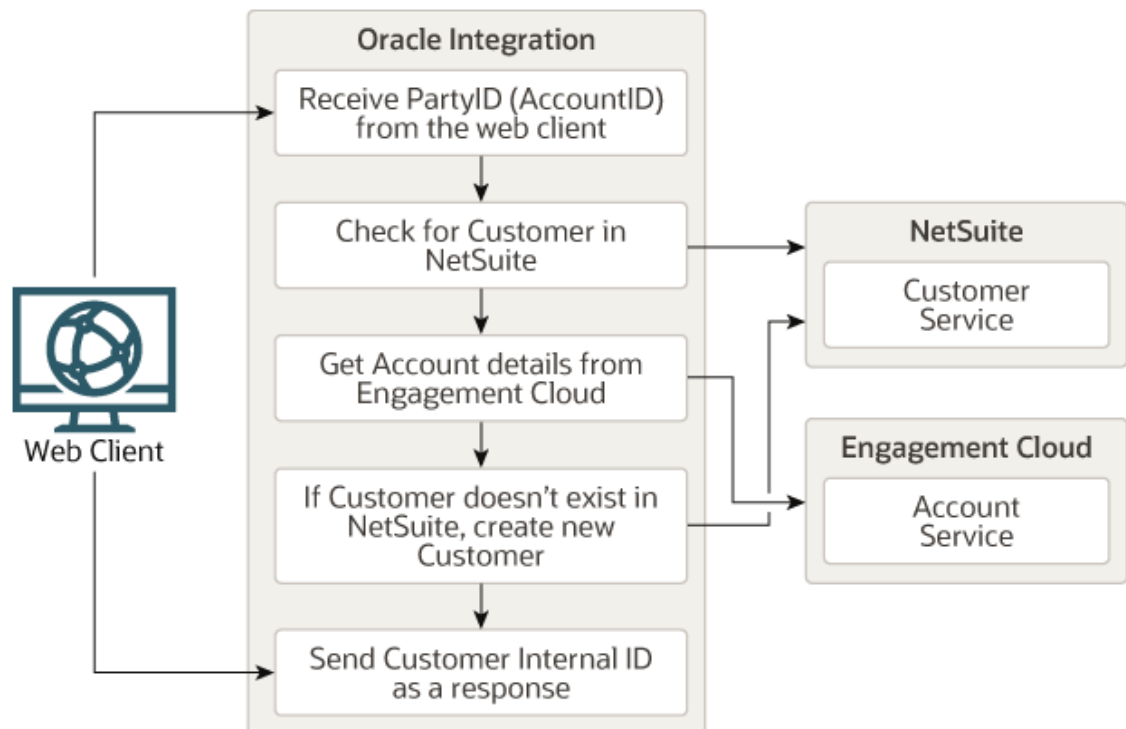
Main Integration Flow - Create Sales Order in NetSuite

This integration flow is triggered when an opportunity is closed with the status *Won* in Oracle CX Sales and B2B Service. It receives the customer and inventory item details associated with the opportunity won, and it calls the subsidiary integration flows to fetch the corresponding customer and inventory item details from Oracle NetSuite. After it receives the details, the main flow creates a new sales order in Oracle NetSuite.



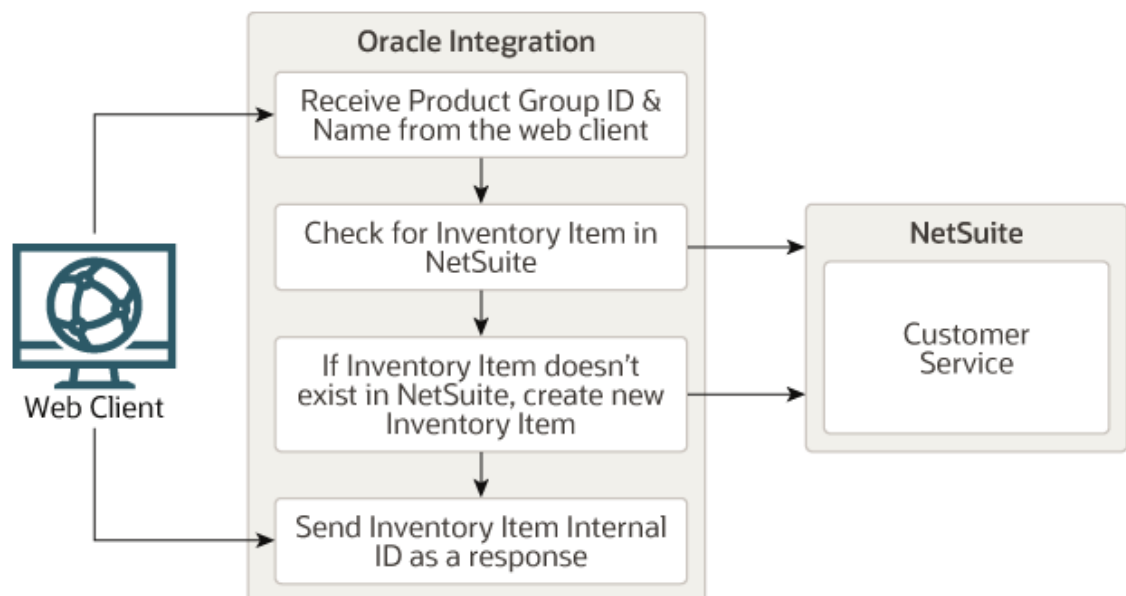
Subsidiary Integration Flow - Fetch Customer Record from NetSuite

This integration flow is triggered by a REST service from the main integration flow. As a parameter, it receives the PartyId (AccountId) of the customer record associated with the opportunity won in Oracle CX Sales and B2B Service. Using this parameter, the flow checks if the corresponding customer record exists in Oracle NetSuite. If the record exists, it returns the internal ID of the record to the main flow. If the record doesn't exist (that is, if it's a new customer), the flow fetches the account details of the customer record from Oracle CX Sales and B2B Service and creates a new customer record in Oracle NetSuite. Finally, it returns the internal ID of this record to the main flow.



Subsidiary Integration Flow - Fetch Inventory Item Record from NetSuite

This integration flow is triggered by a REST service from the main integration flow. As parameters, it receives `prodgroupid` and `prodgroupname` of the inventory item associated with the opportunity won in Oracle CX Sales and B2B Service. Using these parameters, the flow checks if the corresponding inventory item record exists in Oracle NetSuite. If the record exists, it returns the internal ID of the record to the main flow. If the record doesn't exist (that is, if it's a new inventory item), the flow creates a new inventory item record in Oracle NetSuite and returns the internal ID of this record to the main flow.



Before You Install the Recipe

You must perform the following configuration tasks on your Oracle CX Sales and B2B Service instance and Oracle NetSuite instance in order to successfully connect to these external systems using Oracle Integration and achieve synchronization.

Configure Oracle CX Sales and B2B Service

To access the Oracle CX Sales and B2B Service instance from Oracle Integration, you'll require a separate user account on Oracle CX Sales and B2B Service. In addition, you must create a Credential Store Framework (CSF) key 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.

1. Create a new user account for Oracle Integration. You'll use the credentials of this user account to connect to Oracle CX Sales and B2B Service from Oracle Integration. For the procedure to create the integration user account, see *Creating an Integration User Account in Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
 - a. Assign the following roles to the user account. For steps to assign roles, see *Assign Integration Roles in Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
 - Integration Specialist
 - AttachmentsUser
 - FND_MANAGE_CATALOG_SERVICE_PRIV
 - b. Make a note of the user name and password you set for the account.
2. Create a CSF key for Oracle Integration.
 - a. Assemble the CSF Key on your Oracle Integration instance and register the same on Oracle SOA Composer. The CSF key entry in the Oracle CX Sales and B2B Service infrastructure stores the Oracle Integration credentials used by Oracle CX Sales and B2B Service. When Oracle Fusion Applications send outbound requests to Oracle Integration (at runtime), these credentials (user name and password) are sent for authentication. For the procedure to create the Oracle Integration CSF key, see *Enabling Event Subscriptions in the Oracle CX Sales and B2B Service in Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*.
 - b. Create the CSF key by providing your Oracle Integration account credentials.
 - c. Ensure that your Oracle Integration account is assigned at least the ServiceInvoker role on the Oracle Integration instance.

Configure Oracle NetSuite

To successfully connect to Oracle NetSuite using Oracle Integration, you must perform certain general configurations on your Oracle NetSuite instance and some configurations specific to the security policy you'll use to access. In this recipe, you'll use the token-based authentication (TBA) security policy.

Log in to your Oracle NetSuite instance as an **Administrator** and execute the following tasks.

1. Perform the general configurations necessary to connect to Oracle NetSuite. See *Enable Features on Oracle NetSuite and Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3*.

2. Perform the TBA-related configuration tasks. See Prerequisites for the Token-Based Authentication Security Policy in *Using the Oracle NetSuite Adapter with Oracle Integration* 3.

Grant the following permissions to the role you'll create in this step.

Tab Name	Permission	Level
Transactions	Adjust Inventory	Full
Transactions	Cash Sale	Full
Transactions	Cash Sale Refund	Full
Transactions	Customer Deposit	Full
Transactions	Customer Payment	Full
Transactions	Customer Refund	Full
Transactions	Fulfill Orders	Full
Transactions	Invoice	Full
Transactions	Item Shipment	Full
Transactions	Sales Order	Full
Transactions	View Payment Events	Full
Reports	Integration	View
Reports	Inventory	View
Reports	Sales	View
Lists	Accounts	Full
Lists	Cases	Full
Lists	Companies	Full
Lists	Contacts	Full
Lists	Custom Record Entries	Full
Lists	Customers	Full
Lists	Employees	Full
Lists	Items	Full
Lists	Perform Search	Full
Setup	REST Web Services	Full
Setup	SOAP Web Services	Full
Setup	User Access Tokens	Full

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle CX Sales and B2B Service Connection

1. In the Connections section, click the connection name.

2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: `https://customer_chosen_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
Username	Enter the user name of the account created for Oracle Integration on the Oracle CX Sales and B2B Service. See Configure Oracle CX Sales and B2B Service .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle NetSuite Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
WSDL URL	Enter the Oracle NetSuite WSDL URL; for example, <code>https://webservices.netsuite.com/wSDL/<NetSuite_application_version>/netsuite.wsdl..</code> where <code><NetSuite_application_version></code> is the version of the Oracle NetSuite application. For example, <code>v2015_1_0</code> . See <i>Assemble the Oracle NetSuite WSDL URL in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Token Based Authentication .
Consumer Key	Enter the consumer key of the integration record in Oracle NetSuite. See <i>Create an Integration Record for Oracle Integration in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Consumer Secret	Enter the consumer secret of the integration record in Oracle NetSuite.
Token	Enter the token ID provided by Oracle NetSuite. See <i>Create an Access Token for the User Account in Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .
Token Secret	Enter the token secret provided by Oracle NetSuite.

Field	Information to Enter
Account ID	Enter your Oracle NetSuite account identifier. See Make a Note of the NetSuite Account ID in <i>Using the Oracle NetSuite Adapter with Oracle Integration 3</i> .

Note

You must enter this information in capital letters.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Configure the Oracle REST Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to the Oracle CX Sales and B2B Service instance using a separate account, for example, as a sales user. Click **Opportunities** on the home page.
 - b. On the Opportunities page, click **Create Opportunity**.
 - c. On the Create Opportunity page:
 - i. Enter a name for the opportunity.
 - ii. Select an account for the opportunity (mandatory).
 - iii. Set a close date, and click **Save and Continue**.
 - d. On the Summary page:
 - i. Select **Won** in the **Status** field. Choose a suitable reason in the **Win/Loss Reason** field.
 - ii. Scroll to the Products section, and click **Add** to add a product group.

- iii. With **Group** selected in the **Type** field, select a product group in the **Name** field. Enter other necessary details for the product group, such as quantity, estimated price, and so on.

Note

For successful execution of the recipe, you must assign only one product group to the opportunity, and you must not add a product item to the opportunity.

- iv. Click **Save and Close**.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Log in to your Oracle NetSuite instance as an Administrator and check for the new sales order, customer, and inventory item records.
 - a. To view the sales order: On the Oracle NetSuite home page, select **Transactions**, then **Sales**, then **Enter Sales Orders**, and then **List**.
 - b. To view the customer record: On the Oracle NetSuite home page, select **Lists**, then **Relationships**, and then **Customers**.
 - c. To view the inventory item record: On the Oracle NetSuite home page, select **Lists**, then **Accounting**, and then **Items**.

Related Documentation

- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
- *Using the Oracle NetSuite Adapter with Oracle Integration 3*

Sync Oracle Service Cloud Incidents with Oracle CPQ Transactions

Use this recipe to synchronize Oracle Service Cloud (RightNow) incidents with Oracle Configure, Price, and Quote Cloud (Oracle CPQ) transactions.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe synchronizes Oracle Service Cloud incidents with Oracle CPQ transactions. To use the recipe, you must install the recipe and configure connections and other resources within the recipe. Subsequently, you can activate the integration flows of the recipe. The recipe contains three integration flows.

When an incident is created in Oracle Service Cloud, the first integration is triggered and it creates a corresponding transaction with required information such as customer details and service agent details in Oracle CPQ. This in turn triggers the second integration that updates the Oracle Service Cloud incident with the transaction Id of the corresponding transaction that

was created in Oracle CPQ. When the incident is updated with the transaction Id and the service agent changes the status of the incident to *Solved* in Oracle Service Cloud, it triggers the third integration which changes the corresponding transaction status to *Canceled* in Oracle CPQ.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Oracle Service Cloud
- Oracle CPQ
- An account on Oracle Service Cloud with the Administrator role
- An account on Oracle CPQ with the Administrator role
- Access to the Oracle Service Cloud Console

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Service Cloud and Oracle CPQ instances in order to successfully connect to these external systems using Oracle Integration and synchronize Oracle Service Cloud incidents with Oracle CPQ transactions.

Configure Oracle Service Cloud

To access Oracle Service Cloud using Oracle Integration and ensure that Oracle Service Cloud incidents and Oracle CPQ transactions are synchronizied, you must perform certain configuration tasks in your Oracle Service Cloud instance.

Log in to your Oracle Service Cloud instance as an **Administrator** and execute the following tasks.

1. Complete the prerequisite tasks for creating a connection between Oracle Service Cloud and Oracle Integration. See Prerequisites for Creating a Connection in *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*.
2. Add or edit an incident custom field.
 - a. In the navigation pane, click **Configuration**.
 - b. Expand **Database**, and double-click **Custom Fields**.
 - c. To create a new incident custom field, click **Incidents** in **Custom Fields**, and then click **New**.
 - d. Alternatively, to edit a specific incident custom field, click it in the tree.
 - e. In the **Name** field, enter the name of the custom field. For example, CPQTransaction.
 - f. Click **Data Type** and select the data type for the field.
3. Add the incident custom field to the Incident page.
 - a. In the navigation pane, click **Configuration**.
 - b. Expand **Application Appearance**, and double-click **Incident workspace**.
 - c. In the Insert Field section, select the incident custom field that you created/edited in step 2, and drag it to the **Incident workspace**.
 - d. Click **Save**.
4. Add fields to a custom object.

- a. In the navigation pane, click **Configuration**.
- b. Expand **Service**, and double-click **Object Designer**.
- c. Select a package to which you want to add the custom object, or create a new package for the object by clicking **New** and selecting **Package**. Enter a name for the package.
- d. Click **Fields** and then click **Add New Field** to add a customized field to the custom object.

For example, you can add the following custom fields:

Field	Type
title	text
office_phone	text
mobile	text
email	text

- e. Click **Deploy**.
5. Store values to the custom fields.
 - a. In the navigation pane, click **Configuration**.
 - b. Expand **Staff Management**, select and double-click **Staff Accounts by Profile**.
 - c. Choose an agent from the list. Double-click the agent who will be responsible for creating incidents for customers.
 - d. Click **Custom Fields**, and enter values for the custom fields (title, office_phone, mobile, and email) that you created.
 - e. Click **Save**.
6. Add products to Oracle Service Cloud in order to ensure that those products are available in Oracle CPQ.
 - a. In the navigation pane, click **Configuration**.
 - b. Expand **Service** and double-click **Products**, then **Categories**, and then **Dispositions**.
 - c. To add products, click **New** and then select **Sibling or Child**.
 - d. Enter the name of the product. Note that the product name must be a combination of part number and name of the product. For example, AS09743-ExampleProduct.
 - e. Set Visibility to **Admins** and click **Save**.

Configure Oracle CPQ

To access Oracle CPQ using Oracle Integration you must perform certain configuration tasks in your Oracle CPQ instance.

Log in to your Oracle CPQ instance as an **Administrator** and execute the following tasks.

1. Create a user account for Oracle Integration with the **Web Services Only** permission.
 - a. On the Oracle CPQ home page, click **Admin Drawer** on the title bar.
 - b. In the Admin navigation pane, expand **Users** and click **Internal Users**.
The User Administration List page appears.
 - c. Add the user from the User Administration List page.

For details, see [Setting Up Users](#).

2. Obtain the REST Catalog URL. See *Prerequisites for Creating a Connection in Using the Oracle CPQ Adapter with Oracle Integration 3*.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle Service Cloud Connection

1. In the Connections section, click the connection name.
 2. In the Properties section, specify the WSDL to use. See [Configure Oracle Service Cloud](#).
 3. In the Security section, enter the username and password that you obtained. See [Configure Oracle Service Cloud](#).
- Note that only the Username Password Token security policy is supported.
4. Click **Save**. If prompted, click **Save** again.
 5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle CPQ Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Select REST Catalog URL . See Configure Oracle CPQ .
Connection URL	Enter the instance URL of Oracle CPQ.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the username of the account created for Oracle Integration in Oracle CPQ. See Configure Oracle CPQ .
Password	Enter the password of the account created for Oracle Integration in Oracle CPQ.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Update Integration Property

You have to update the integration property value so that Oracle CPQ quote approval notification messages are sent to the right Slack channel.

1. In the Integrations section, click **Actions** ... on the integration flow, then select **Update property values**.
2. In the Update Property Values dialog, click **pricebookname**.
3. In the **New Value** field, enter the name of the pricebook. For example, `default_price_book`.
4. Click **Submit**.

A message confirms that the integration property has been updated successfully.

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to the Oracle Service Cloud instance.
 - b. As the Service agent, create an incident with customer details.

You have now successfully triggered the recipe.
 - c. Refresh the Oracle Service Cloud incident and check if it's updated with the Transaction Id.
 - d. If the incident is updated with the Transaction Id, change the status of the Oracle Service Cloud incident to *Solved*. From the drop-down list of the incident's **Status** field, select **Solved**.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if the corresponding transaction is canceled in Oracle CPQ.
 - a. Log in to the Oracle CPQ instance.
 - b. Navigate to the transaction that was created corresponding to the Service Cloud incident.
 - c. Check if the status of the transaction is *Canceled*.

Related Documentation

- *Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
- *Using the Oracle CPQ Adapter with Oracle Integration 3*

Validate ID for Loan Application Using OCI AI Services

Use this recipe to validate ID for loan application using the following OCI AI services - Oracle Cloud Infrastructure Vision (OCI Vision) and Oracle Cloud Infrastructure Generative AI (OCI Generative AI).

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe downloads a loan application from a Gmail account, and uses OCI Vision to validate the loan application ID and the applicant's photo. Then uses OCI Generative AI to write an email with the results.

To use the recipe, you must install the recipe and configure the connections and other resources within it. The main integration runs on a schedule and fetches the loan application from a Gmail account. Then, it sends the loan application to the child integration for processing. The child integration uses OCI Vision to validate the ID and the applicant's photo in the loan application. Then, it uses OCI Generative AI to write an email with the results and notify the applicant.

System and Access Requirements

- Oracle Integration 3
- OCI Vision
- OCI Generative AI
- A valid account on Google Gmail

Before You Install the Recipe

You must perform the following configuration tasks on Gmail, OCI Vision and OCI Generative AI in order to successfully validate ID for loan application.

Configure Gmail

You must perform the general configuration tasks for creating a connection. See Prerequisites for Creating a Connection in *Using the Google Gmail Adapter with Oracle Integration 3*.

Configure OCI Vision

Configure and apply policies at the compartment level for OCI Vision users.

See [Policy to Grant Users Access to Vision APIs](#) in *OCI Vision* documentation.

Ensure that you select the same OCI compartment (where you applied the OCI Vision policies) when you [configure the integration flow](#) of the recipe to use the **OCI Vision** action.

Configure OCI Generative AI

Configure and apply policies at the compartment level for OCI Generative AI users.

See [Getting Access to Generative AI](#) in *OCI Generative AI* documentation.

Ensure that you select the same OCI compartment (where you applied the OCI Generative AI policies) when you [configure the integration flow](#) of the recipe to use the **OCI Generative AI** action.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Gmail Connection

1. In the Connections section, click the connection name.
2. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	By default Google OAuth Authorization Code Credentials is selected.
Client ID	Enter the client ID.
Client Secret	Enter the client secret.
Scope	Select a scope from the drop-down list. For example, <code>https://mail.google.com/</code>

3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
5. To return to the project workspace, click **Go back** .

Configure the REST Trigger Connection

1. In the Connections section, click the connection name.
2. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Configure the Integration Flow

Edit the **Process Application** integration flow to configure actions for the following OCI AI services.

- [OCI Vision](#)
- [OCI Generative AI](#)

OCI Vision

1. In the Integrations section, click the integration flow name.
2. On the integration canvas, click the **OCI Vision** element, and then click **Actions ...** and select **Edit**.
3. Click **Next** and select the OCI compartment.
4. Click **Next** and then **Done**.
5. Click **Save** to save the integration.
6. To return to the project workspace, click **Go back** .

To know more about the vision action, see Analyze and Extract Information from Images with a Vision Action in *Using Integrations in Oracle Integration 3*.

OCI Generative AI

1. In the Integrations section, click the integration flow name.
2. On the integration canvas, click the **OCI Generative AI** element, and then click **Actions ...** and select **Edit**.
3. Click **Next** and select the OCI region, OCI compartment, model, and model ID.
4. Click **Next** and then **Done**.
5. Click **Save** to save the integration.
6. To return to the project workspace, click **Go back** .

To know more about the generative AI action, see Generate Text with a Generative AI Action in *Using Integrations in Oracle Integration 3*.

Configure the Lookup Tables

Edit the lookup tables and configure them as necessary.

AI_Properties

1. In the Lookups section, click the lookup name.
2. Update the value for the **Ref_Number_Pattern** lookup key that would be used by OCI Vision to validate the loan application ID.
3. Update the values for the following AI prompt lookup keys that would be used by OCI Generative AI to compose the loan application validation email.
 - **InvalidPhoto_Prompt**
 - **UnclearPhoto_Prompt**
 - **GeneralError_Prompt**

- **ValidApplicationPrompt**
- 4. Click **Save**. If prompted, click **Save** again.
- 5. To return to the project workspace, click **Go back** .

Email_Properties

1. In the Lookups section, click the lookup name.
2. Edit the values of the following lookup keys related to email properties.

Key	Value
to	Enter the email address to which error notifications have to be sent.
from	Enter the email address from which error notifications have to be sent.
validapplication_from	Enter the email address from which notification emails about loan application validation are to be sent
validapplication_to	Enter the email address of the loan applicant to whom notification about loan application ID validation have to be sent.
gmail_subject	Specify the email subject that would be used by the recipe to filter out emails (in the Gmail account) from which loan applications are to be fetched for validation.

3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. In the Integrations section of the project workspace, click **Actions**  on the **Fetch Applications from Gmail** integration flow, then select **Run**

The Configure and run page is displayed, where you can specify a value for the `lastRun` parameter. This parameter stores the date of the most-recent successful run of the integration flow. For the initial run, it contains a default value. The parameter's value is automatically updated after each successful run, and only the loan application's received after the date stored as the parameter's **Current Value** are processed by the integration in each run. If you want to change the date for a specific scenario, enter the date of your choice in the **New Value** field in the format, yyyy-MM-ddTHH:mm:ss.
 - b. On the Configure and run page, click **Run**.

You've now successfully submitted the integration for execution.

Note

You can also schedule this integration to run at a date, time, and frequency of your choosing. See [Define the Integration Schedule](#).

3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if the loan applicant received an email about the loan application validation status.

Related Documentation

- [Using the Google Gmail Adapter with Oracle Integration 3](#)
- [Using the REST Adapter with Oracle Integration 3](#)
- [OCI Vision](#)
- [OCI Generative AI](#)

View Orders from Oracle ERP Cloud in Oracle CX Sales and B2B Service

Use this recipe to view details of an order from Oracle ERP Cloud in Oracle CX Sales and B2B Service.

Topics:

- [About This Recipe](#)
- [Before You Install the Recipe](#)
- [Install and Configure the Recipe](#)
- [Activate and Run the Recipe](#)

About This Recipe

This recipe creates a custom object `OrderManagementPOC` in Oracle CX Sales and B2B Service when an order is created in Oracle ERP Cloud. The details of the order created can be viewed in the Oracle CX Sales and B2B Service custom object in real time.

To use the recipe, you must install the recipe and configure connections and other resources within the recipe. When an order is created in Oracle ERP Cloud, the integration flow of the recipe is triggered, and the recipe creates a custom object `OrderManagementPOC` in Oracle CX Sales and B2B Service. It is assumed that the Oracle ERP Cloud Account is already synced with the Oracle CX Sales and B2B Service Account and the custom object `OrderManagementPOC` is created as child object of Account. The details (such as Item Name, Item Description, Order Status, Quantity Purchased, and so on) of the order created in Oracle ERP Cloud can be viewed real time on the `OrderManagementPOC` custom object created in Oracle CX Sales and B2B Service.

System and Access Requirements

- Oracle Integration, Version 21.2.1 or higher
- Oracle ERP Cloud
- Oracle CX Sales and B2B Service

- An account on Oracle ERP Cloud with the Administrator role
- An account on Oracle CX Sales and B2B Service with the Administrator role

Before You Install the Recipe

You must perform the following configuration tasks on your Oracle ERP Cloud instance and Oracle CX Sales and B2B Service instance in order to successfully connect to these external systems using Oracle Integration.

Configure Oracle ERP Cloud

To access the Oracle ERP Cloud instance from Oracle Integration, you'll require a separate user account on Oracle ERP Cloud.

Log in to your Oracle ERP Cloud instance as an **Administrator** and perform the following tasks.

1. Create a user account for Oracle Integration. Make a note of the user name and password you set for the account. You'll use the credentials of this user account to connect to Oracle ERP Cloud from Oracle Integration.
2. Assign the following roles to the user account. For more information, see *Assign Required Roles to an Integration User in Using the Oracle ERP Cloud Adapter with Oracle Integration 3*.
 - Integration Specialist
 - Oracle ERP Cloud-specific data access to the integration user
3. Enable the business event Sales Order Notification.
See *Supported SCM and Procurement Business Events*.
For information on subscribing to events in Oracle ERP Cloud, see *Configure Oracle ERP Cloud for Event Subscriptions*.

Configure Oracle CX Sales and B2B Service

To access the Oracle CX Sales and B2B Service instance from Oracle CX Sales and B2B Service, you'll have to create a separate user account on Oracle CX Sales and B2B Service, and assign the required roles to the user account.

In addition, perform the following tasks to configure Oracle CX Sales and B2B Service for this recipe.

1. Create the OrderManagementPOC custom object as child object of Account with necessary fields.
 - a. Create OrderManagementPOC custom object.
 - i. In the Navigator, select **Application Composer**.
 - ii. In the Custom Objects list, click **Create a New Object**.
 - iii. In the Create Custom Object dialog, enter the object information (OrderManagementPOC), and then click **OK**.
The custom object appears in the Custom Objects list in the Objects menu.
 - b. Add fields to the OrderManagementPOC custom object.

- i. In the Objects menu, expand the OrderManagementPOC custom object's node in the Custom Objects list.
 - ii. Click **Fields**.
 - iii. On the Fields page, click **Create a custom field**.
 - iv. In the Select Field Type dialog, select the appropriate field type, and then click **OK**.
 - v. Repeat the above steps to create as many fields as required.
2. Ensure Registry ID value of account object in Oracle CX Sales and B2B Service matches with BuyingPartyNumber of order object in Oracle ERP Cloud.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

Install the recipe on your Oracle Integration instance. See [Install a Recipe or Accelerator](#).

After you've installed the recipe, configure the following resources:

Configure the Oracle ERP Cloud Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle ERP Cloud host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on the Oracle ERP Cloud instance. See Configure Oracle ERP Cloud .
Password	Enter the password of the account created for Oracle Integration on the Oracle ERP Cloud instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

Configure the Oracle CX Sales and B2B Service Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the Oracle CX Sales and B2B Service host name. For example: `https://your_domain_name.fa.DC.oraclecloud.com`.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select Username Password Token .
User Name	Enter the user name of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance. See Configure Oracle CX Sales and B2B Service .
Password	Enter the password of the account created for Oracle Integration on the Oracle CX Sales and B2B Service instance.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

Activate and Run the Recipe

After you've configured the connections, you can activate and run the recipe.

1. Activate the recipe. See [Activate a Recipe or Accelerator](#).
2. Run the recipe.
 - a. Log in to your Oracle ERP Cloud instance.
 - b. Open the Order Management module.
 - c. Create an order by selecting the Account that is synced to Oracle CX Sales and B2B Service.
 - d. Click **Submit**.

You've now successfully triggered the recipe.
3. Monitor the running of the integration flow in Oracle Integration. See [Monitor Integrations](#).
4. Check if you can view in Oracle CX Sales and B2B Service the details of the order that was created in Oracle ERP Cloud.
 - a. Log in to your Oracle CX Sales and B2B Service instance.
 - b. Open the OrderManagementPOC custom object that was created as child object of Account.
 - c. Check if you can view the details of the order that was created in Oracle ERP Cloud on the OrderManagementPOC custom object.

Related Documentation

- *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- *Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*